

IEEE Std 1003.1-2001
(Revision of IEEE Std 1003.1-1996
and IEEE Std 1003.2-1992)

**Open Group Technical Standard
Base Specifications, Issue 6**

1003.1TM

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

Shell and Utilities, Issue 6

Approved 12 September 2001
The Open Group

IEEE Sponsor
Portable Applications Standards Committee
of the
IEEE Computer Society

Approved 6 December 2001
IEEE-SA Standards Board



THE *Open* GROUP

Abstract

This standard defines a standard operating system interface and environment, including a command interpreter (or “shell”), and common utility programs to support applications portability at the source code level. It is the single common revision to IEEE Std 1003.1-1996, IEEE Std 1003.2-1992, and the Base Specifications of The Open Group Single UNIX®† Specification, Version 2. This standard is intended to be used by both applications developers and system implementors and comprises four major components (each in an associated volume):

- General terms, concepts, and interfaces common to all volumes of this standard, including utility conventions and C-language header definitions, are included in the Base Definitions volume.
- Definitions for system service functions and subroutines, language-specific system services for the C programming language, function issues, including portability, error handling, and error recovery, are included in the System Interfaces volume.
- Definitions for a standard source code-level interface to command interpretation services (a “shell”) and common utility programs for application programs are included in the Shell and Utilities volume.
- Extended rationale that did not fit well into the rest of the document structure, containing historical information concerning the contents of this standard and why features were included or discarded by the standard developers, is included in the Rationale (Informative) volume.

The following areas are outside the scope of this standard:

- Graphics interfaces
- Database management system interfaces
- Record I/O considerations
- Object or binary code portability
- System configuration and resource availability

This standard describes the external characteristics and facilities that are of importance to applications developers, rather than the internal construction techniques employed to achieve these capabilities. Special emphasis is placed on those functions and facilities that are needed in a wide variety of commercial applications.

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)

† See **Trademarks** (on page xxvi).

Copyright © 2001 by the Institute of Electrical and Electronics Engineers, Inc. and The Open Group

Shell and Utilities, Issue 6

Published 6 December 2001 by the Institute of Electrical and Electronics Engineers, Inc.

3 Park Avenue, New York, NY 10016-5997, U.S.A.

ISBN: 0-7381-3094-4 PDF 0-7381-3010-9/SS94956 CD-ROM 0-7381-3129-6/SE94956

Printed in the United States of America by the IEEE.

Published 6 December 2001 by The Open Group

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

Document Number: C952

ISBN: U.K. 1-85912-257-4 U.S. 1-931624-09-7

Printed in the U.K. by The Open Group.

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 <http://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.

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.

The IEEE and its designees are the sole entities that may authorize the use of IEEE-owned certification marks and/or trademarks to indicate compliance with the materials set forth herein. 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>.

Full catalog and ordering information on all IEEE publications is available from the IEEE Online Catalog & Store at <http://shop.ieee.org/store>.

1. For this standard, please send comments via the Austin Group as requested on page iii.

2. Please refer to the special provisions for this standard on page iii 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, a vendor and technology-neutral consortium, is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost, and risks associated with integrating new technology across the enterprise.

The Open Group's mission is to offer all organizations concerned with open information infrastructures a forum to share knowledge, integrate open initiatives, and certify approved products and processes in a manner in which they continue to trust our impartiality.

In the global eCommerce world of today, no single economic entity can achieve independence while still ensuring interoperability. The assurance that products will interoperate with each other across differing systems and platforms is essential to the success of eCommerce and business workflow. The Open Group, with its proven testing and certification program, is the international guarantor of interoperability in the new century.

The Open Group provides opportunities to exchange information and shape the future of IT. The Open Group's members include some of the largest and most influential organizations in the world. The flexible structure of The Open Groups membership allows for almost any organization, no matter what their size, to join and have a voice in shaping the future of the IT world.

More information is available on The Open Group web site at <http://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 <http://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 <http://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 <http://www.opengroup.org/corrigenda>.

Full catalog and ordering information on all Open Group publications is available at <http://www.opengroup.org/pubs>.

Contents

Chapter	1	Introduction.....	1
1.1	Scope.....	1	
1.2	Conformance	1	
1.3	Normative References	1	
1.4	Change History	1	
1.5	Terminology	1	
1.6	Definitions	3	
1.7	Relationship to Other Documents.....	3	
1.7.1	System Interfaces	3	
1.7.1.1	Process Attributes.....	3	
1.7.1.2	Concurrent Execution of Processes.....	3	
1.7.1.3	File Access Permissions	4	
1.7.1.4	File Read, Write, and Creation	4	
1.7.1.5	File Removal	6	
1.7.1.6	File Time Values	6	
1.7.1.7	File Contents	6	
1.7.1.8	Pathname Resolution.....	7	
1.7.1.9	Changing the Current Working Directory.....	7	
1.7.1.10	Establish the Locale	7	
1.7.1.11	Actions Equivalent to Functions.....	7	
1.7.2	Concepts Derived from the ISO C Standard.....	7	
1.7.2.1	Arithmetic Precision and Operations	7	
1.7.2.2	Mathematical Functions	9	
1.8	Portability	9	
1.8.1	Codes.....	9	
1.9	Utility Limits.....	17	
1.10	Grammar Conventions	19	
1.11	Utility Description Defaults.....	20	
1.12	Considerations for Utilities in Support of Files of Arbitrary Size ...	27	
1.13	Built-In Utilities.....	28	
Chapter	2	Shell Command Language	29
2.1	Shell Introduction	29	
2.2	Quoting.....	30	
2.2.1	Escape Character (Backslash).....	30	
2.2.2	Single-Quotes.....	30	
2.2.3	Double-Quotes	30	
2.3	Token Recognition.....	31	
2.3.1	Alias Substitution	32	
2.4	Reserved Words	33	
2.5	Parameters and Variables.....	33	
2.5.1	Positional Parameters.....	33	

2.5.2	Special Parameters.....	34
2.5.3	Shell Variables.....	34
2.6	Word Expansions.....	36
2.6.1	Tilde Expansion.....	37
2.6.2	Parameter Expansion	37
2.6.3	Command Substitution.....	40
2.6.4	Arithmetic Expansion	41
2.6.5	Field Splitting.....	42
2.6.6	Pathname Expansion.....	42
2.6.7	Quote Removal.....	42
2.7	Redirection	43
2.7.1	Redirecting Input.....	44
2.7.2	Redirecting Output.....	44
2.7.3	Appending Redirected Output	44
2.7.4	Here-Document.....	44
2.7.5	Duplicating an Input File Descriptor.....	45
2.7.6	Duplicating an Output File Descriptor	45
2.7.7	Open File Descriptors for Reading and Writing.....	46
2.8	Exit Status and Errors	46
2.8.1	Consequences of Shell Errors	46
2.8.2	Exit Status for Commands	46
2.9	Shell Commands	47
2.9.1	Simple Commands	47
2.9.1.1	Command Search and Execution	48
2.9.2	Pipelines.....	49
2.9.3	Lists.....	50
2.9.3.1	Asynchronous Lists.....	50
2.9.3.2	Sequential Lists.....	51
2.9.3.3	AND Lists.....	51
2.9.3.4	OR Lists.....	51
2.9.4	Compound Commands.....	52
2.9.4.1	Grouping Commands.....	52
2.9.4.2	The for Loop.....	52
2.9.4.3	Case Conditional Construct	53
2.9.4.4	The if Conditional Construct.....	53
2.9.4.5	The while Loop.....	54
2.9.4.6	The until Loop	54
2.9.5	Function Definition Command.....	54
2.10	Shell Grammar.....	55
2.10.1	Shell Grammar Lexical Conventions.....	55
2.10.2	Shell Grammar Rules	56
2.11	Signals and Error Handling	61
2.12	Shell Execution Environment.....	61
2.13	Pattern Matching Notation	62
2.13.1	Patterns Matching a Single Character	62
2.13.2	Patterns Matching Multiple Characters	63
2.13.3	Patterns Used for Filename Expansion	63
2.14	Special Built-In Utilities.....	64

<i>break</i>	65
<i>colon</i>	67
<i>continue</i>	69
<i>dot</i>	71
<i>eval</i>	73
<i>exec</i>	75
<i>exit</i>	77
<i>export</i>	79
<i>readonly</i>	81
<i>return</i>	83
<i>set</i>	85
<i>shift</i>	91
<i>times</i>	93
<i>trap</i>	95
<i>unset</i>	98
Chapter 3	
Batch Environment Services	101
3.1 General Concepts.....	101
Batch Client-Server Interaction.....	101
3.1.2 Batch Queues	101
3.1.3 Batch Job Creation	102
3.1.4 Batch Job Tracking	102
3.1.5 Batch Job Routing	102
3.1.6 Batch Job Execution.....	103
3.1.7 Batch Job Exit.....	103
3.1.8 Batch Job Abort	103
3.1.9 Batch Authorization	103
3.1.10 Batch Administration.....	104
3.1.11 Batch Notification	104
3.2 Batch Services	104
3.2.1 Batch Job States	105
3.2.2 Deferred Batch Services.....	106
3.2.2.1 Batch Job Execution.....	106
3.2.2.2 Batch Job Routing	113
3.2.2.3 Batch Job Exit.....	113
3.2.2.4 Batch Server Restart	114
3.2.2.5 Batch Job Abort	114
3.2.3 Requested Batch Services	115
3.2.3.1 Delete Batch Job Request.....	115
3.2.3.2 Hold Batch Job Request.....	116
3.2.3.3 Batch Job Message Request.....	116
3.2.3.4 Batch Job Status Request	117
3.2.3.5 Locate Batch Job Request	117
3.2.3.6 Modify Batch Job Request.....	117
3.2.3.7 Move Batch Job Request.....	118
3.2.3.8 Queue Batch Job Request	118
3.2.3.9 Batch Queue Status Request.....	119
3.2.3.10 Release Batch Job Request.....	119

3.2.3.11	Rerun Batch Job Request	120
3.2.3.12	Select Batch Jobs Request.....	120
3.2.3.13	Server Shutdown Request.....	120
3.2.3.14	Server Status Request.....	121
3.2.3.15	Signal Batch Job Request	121
3.2.3.16	Track Batch Job Request	121
3.3	Common Behavior for Batch Environment Utilities	122
3.3.1	Batch Job Identifier	122
3.3.2	Destination	123
3.3.3	Multiple Keyword-Value Pairs	123
Chapter 4	Utilities.....	125
	Index.....	1079

List of Figures

4-1	pax Format Archive Example.....	710
-----	---------------------------------	-----

List of Tables

1-1	Actions when Creating a File that Already Exists	5
1-2	Selected ISO C Standard Operators and Control Flow Keywords.....	8
1-3	Utility Limit Minimum Values.....	17
1-4	Symbolic Utility Limits.....	18
1-5	Regular Built-In Utilities	28
3-1	Batch Utilities.....	101
3-2	Environment Variable Summary	105
3-3	Next State Table	107
3-4	Results/Output Table	108
3-5	Batch Services Summary	115
4-1	Expressions in Decreasing Precedence in <i>awk</i>	156
4-2	Escape Sequences in <i>awk</i>	162
4-3	Operators in <i>bc</i>	198
4-4	Programming Environments: Type Sizes	215
4-5	Programming Environments: <i>c99</i> and <i>cc</i> Arguments.....	216
4-6	ASCII to EBCDIC Conversion.....	303
4-7	ASCII to IBM EBCDIC Conversion.....	304
4-8	File Utility Output Strings	444
4-9	Table Size Declarations in <i>lex</i>	535
4-10	Escape Sequences in <i>lex</i>	537
4-11	ERE Precedence in <i>lex</i>	538
4-12	Named Characters in <i>od</i>	679
4-13	ustar Header Block	715
4-14	ustar <i>mode</i> Field	716
4-15	Octet-Oriented <i>cpio</i> Archive Entry.....	718
4-16	Values for <i>cpio c_mode</i> Field	719
4-17	Variable Names and Default Headers in <i>ps</i>	752

Contents

4-18	Environment Variable Values (Utilities)	801
4-19	Control Character Names in <i>stty</i>	886
4-20	Circumflex Control Characters in <i>stty</i>	886
4-21	uuencode Base64 Values.....	970
4-22	Internal Limits in <i>yacc</i>	1072

Introduction

Note: This introduction is not part of IEEE Std 1003.1-2001, Standard for Information Technology — Portable Operating System Interface (POSIX).

This standard has been jointly developed by the IEEE and The Open Group. It is both an IEEE Standard and an Open Group Technical Standard.

The Austin Group

This standard was developed, and is maintained, by a joint working group of members of the IEEE Portable Applications Standards Committee, members of The Open Group, and members of ISO/IEC Joint Technical Committee 1. This joint working group is known as the Austin Group.³ The Austin Group arose out of discussions amongst the parties which started in early 1998, leading to an initial meeting and formation of the group in September 1998. The purpose of the Austin Group has been to revise, combine, and update the following standards: ISO/IEC 9945-1, ISO/IEC 9945-2, IEEE Std 1003.1, IEEE Std 1003.2, and the Base Specifications of The Open Group Single UNIX Specification.

After two initial meetings, an agreement was signed in July 1999 between The Open Group and the Institute of Electrical and Electronics Engineers (IEEE), Inc., to formalize the project with the first draft of the revised specifications being made available at the same time. Under this agreement, The Open Group and IEEE agreed to share joint copyright of the resulting work. The Open Group has provided the chair and secretariat for the Austin Group.

The base document for the revision was The Open Group's Base volumes of its Single UNIX Specification, Version 2. These were selected since they were a superset of the existing POSIX.1 and POSIX.2 specifications and had some organizational aspects that would benefit the audience for the new revision.

The approach to specification development has been one of “write once, adopt everywhere”, with the deliverables being a set of specifications that carry the IEEE POSIX designation and The Open Group's Technical Standard designation, and, if approved, an ISO/IEC designation. This set of specifications forms the core of the Single UNIX Specification, Version 3.

This unique development has combined both the industry-led efforts and the formal standardization activities into a single initiative, and included a wide spectrum of participants. The Austin Group continues as the maintenance body for this document.

Anyone wishing to participate in the Austin Group should contact the chair with their request. There are no fees for participation or membership. You may participate as an observer or as a contributor. You do not have to attend face-to-face meetings to participate; electronic participation is most welcome. For more information on the Austin Group and how to participate, see <http://www.opengroup.org/austin>.

3. The Austin Group is named after the location of the inaugural meeting held at the IBM facility in Austin, Texas in September 1998.

Background

The developers of this standard represent a cross section of hardware manufacturers, vendors of operating systems and other software development tools, software designers, consultants, academics, authors, applications programmers, and others.

Conceptually, this standard describes a set of fundamental services needed for the efficient construction of application programs. Access to these services has been provided by defining an interface, using the C programming language, a command interpreter, and common utility programs that establish standard semantics and syntax. Since this interface enables application writers to write portable applications—it was developed with that goal in mind—it has been designated POSIX,⁴ an acronym for Portable Operating System Interface.

Although originated to refer to the original IEEE Std 1003.1-1988, the name POSIX more correctly refers to a *family* of related standards: IEEE Std 1003.n and the parts of ISO/IEC 9945. In earlier editions of the IEEE standard, the term POSIX was used as a synonym for IEEE Std 1003.1-1988. A preferred term, POSIX.1, emerged. This maintained the advantages of readability of the symbol “POSIX” without being ambiguous with the POSIX family of standards.

Audience

The intended audience for this standard is all persons concerned with an industry-wide standard operating system based on the UNIX system. This includes at least four groups of people:

1. Persons buying hardware and software systems
2. Persons managing companies that are deciding on future corporate computing directions
3. Persons implementing operating systems, and especially
4. Persons developing applications where portability is an objective

Purpose

Several principles guided the development of this standard:

- Application-Oriented

The basic goal was to promote portability of application programs across UNIX system environments by developing a clear, consistent, and unambiguous standard for the interface specification of a portable operating system based on the UNIX system documentation. This standard codifies the common, existing definition of the UNIX system.

- Interface, Not Implementation

This standard defines an interface, not an implementation. No distinction is made between library functions and system calls; both are referred to as functions. No details of the implementation of any function are given (although historical practice is sometimes indicated in the RATIONALE section). Symbolic names are given for constants (such as signals and error numbers) rather than numbers.

4. The name POSIX was suggested by Richard Stallman. It is expected to be pronounced *pahz-icks*, as in *positive*, not *poh-six*, or other variations. The pronunciation has been published in an attempt to promulgate a standardized way of referring to a standard operating system interface.

- **Source, Not Object, Portability**

This standard has been written so that a program written and translated for execution on one conforming implementation may also be translated for execution on another conforming implementation. This standard does not guarantee that executable (object or binary) code will execute under a different conforming implementation than that for which it was translated, even if the underlying hardware is identical.

- **The C Language**

The system interfaces and header definitions are written in terms of the standard C language as specified in the ISO C standard.

- **No Superuser, No System Administration**

There was no intention to specify all aspects of an operating system. System administration facilities and functions are excluded from this standard, and functions usable only by the superuser have not been included. Still, an implementation of the standard interface may also implement features not in this standard. This standard is also not concerned with hardware constraints or system maintenance.

- **Minimal Interface, Minimally Defined**

In keeping with the historical design principles of the UNIX system, the mandatory core facilities of this standard have been kept as minimal as possible. Additional capabilities have been added as optional extensions.

- **Broadly Implementable**

The developers of this standard endeavored to make all specified functions implementable across a wide range of existing and potential systems, including:

1. All of the current major systems that are ultimately derived from the original UNIX system code (Version 7 or later)
2. Compatible systems that are not derived from the original UNIX system code
3. Emulations hosted on entirely different operating systems
4. Networked systems
5. Distributed systems
6. Systems running on a broad range of hardware

No direct references to this goal appear in this standard, but some results of it are mentioned in the Rationale (Informative) volume.

- **Minimal Changes to Historical Implementations**

When the original version of IEEE Std 1003.1 was published, there were no known historical implementations that did not have to change. However, there was a broad consensus on a set of functions, types, definitions, and concepts that formed an interface that was common to most historical implementations.

The adoption of the 1988 and 1990 IEEE system interface standards, the 1992 IEEE shell and utilities standard, the various Open Group (formerly X/Open) specifications, and the subsequent revisions and addenda to all of them have consolidated this consensus, and this revision reflects the significantly increased level of consensus arrived at since the original versions. The earlier standards and their modifications specified a number of areas where consensus had not been reached before, and these are now reflected in this revision. The authors of the original versions tried, as much as possible, to follow the principles below

when creating new specifications:

1. By standardizing an interface like one in an historical implementation; for example, directories
2. By specifying an interface that is readily implementable in terms of, and backwards-compatible with, historical implementations, such as the extended *tar* format defined in the *pax* utility
3. By specifying an interface that, when added to an historical implementation, will not conflict with it; for example, the *sigaction()* function

This revision tries to minimize the number of changes required to implementations which conform to the earlier versions of the approved standards to bring them into conformance with the current standard. Specifically, the scope of this work excluded doing any ‘‘new’’ work, but rather collecting into a single document what had been spread across a number of documents, and presenting it in what had been proven in practice to be a more effective way. Some changes to prior conforming implementations were unavoidable, primarily as a consequence of resolving conflicts found in prior revisions, or which became apparent when bringing the various pieces together.

However, since it references the 1999 version of the ISO C standard, and no longer supports ‘‘Common Usage C’’, there are a number of unavoidable changes. Applications portability is similarly affected.

This standard is specifically not a codification of a particular vendor’s product.

It should be noted that implementations will have different kinds of extensions. Some will reflect ‘‘historical usage’’ and will be preserved for execution of pre-existing applications. These functions should be considered ‘‘obsolescent’’ and the standard functions used for new applications. Some extensions will represent functions beyond the scope of this standard. These need to be used with careful management to be able to adapt to future extensions of this standard and/or port to implementations that provide these services in a different manner.

- Minimal Changes to Existing Application Code

A goal of this standard was to minimize additional work for the developers of applications. However, because every known historical implementation will have to change at least slightly to conform, some applications will have to change.

This Standard

This standard defines the Portable Operating System Interface (POSIX) requirements and consists of the following volumes:

- Base Definitions
- Shell and Utilities (this volume)
- System Interfaces
- Rationale (Informative)

This Volume

The Shell and Utilities volume describes the commands and utilities offered to application programs on POSIX-conformant systems. Readers are expected to be familiar with the Base Definitions volume.

This volume is structured as follows:

- Chapter 1 explains the status of this volume and its relationship to other formal standards. It also describes the defaults used by the utility descriptions in Chapter 4.
- Chapter 2 describes the command language used in POSIX-conformant systems.
- Chapter 4 consists of reference pages for all utilities available on POSIX-conformant systems.

Comprehensive references are available in the index.

Typographical Conventions

The following typographical conventions are used throughout this standard. In the text, this standard is referred to as IEEE Std 1003.1-2001, which is technically identical to The Open Group Base Specifications, Issue 6.

The typographical conventions listed here are for ease of reading only. Editorial inconsistencies in the use of typography are unintentional and have no normative meaning in this standard.

Reference	Example	Notes
C-Language Data Structure	aiocb	
C-Language Data Structure Member	<i>aio_lio_opcode</i>	
C-Language Data Type	long	
C-Language External Variable	<i>errno</i>	
C-Language Function	<i>system()</i>	
C-Language Function Argument	<i>arg1</i>	
C-Language Function Family	<i>exec</i>	
C-Language Header	<sys/stat.h>	
C-Language Keyword	return	
C-Language Macro with Argument	<i>assert()</i>	
C-Language Macro with No Argument	INET_ADDRSTRLEN	
C-Language Preprocessing Directive	#define	
Commands within a Utility	a, c	
Conversion Specification, Specifier / Modifier Character	<i>%A, g, E</i>	1
Environment Variable	PATH	
Error Number	<i>[EINTR]</i>	
Example Output	Hello, World	
Filename	/tmp	
Literal Character	<i>'c', '\r', '\ '</i>	2
Literal String	<i>"abcde"</i>	2
Optional Items in Utility Syntax	[]	
Parameter	<i><directory pathname></i>	
Special Character	<i><newline></i>	3
Symbolic Constant	_POSIX_VDISABLE	
Symbolic Limit, Configuration Value	{LINE_MAX}	4

Reference	Example	Notes
Syntax	#include <sys/stat.h>	
User Input and Example Code	echo Hello, World	
Utility Name	<i>awk</i>	
Utility Operand	<i>file_name</i>	
Utility Option	-c	
Utility Option with Option-Argument	-w <i>width</i>	

Notes:

1. Conversion specifications, specifier characters, and modifier characters are used primarily in date-related functions and utilities and the *fprintf* and *fscanf* formatting functions.
2. Unless otherwise noted, the quotes shall not be used as input or output. When used in a list item, the quotes are omitted. For literal characters, '\\' (or any of the other sequences such as ''') is the same as the C constant '\\\\' (or '\\''').
3. The style selected for some of the special characters, such as <newline>, matches the form of the input given to the *localedef* utility. Generally, the characters selected for this special treatment are those that are not visually distinct, such as the control characters <tab> or <newline>.
4. Names surrounded by braces represent symbolic limits or configuration values which may be declared in appropriate headers by means of the C #define construct.
5. Brackets shown in this font, "[]", are part of the syntax and do *not* indicate optional items. In syntax the '| |' symbol is used to separate alternatives, and ellipses (" . . . ") are used to show that additional arguments are optional.

Shading is used to identify extensions and options; see Section 1.8.1 (on page 9).

Footnotes and notes within the body of the normative text are for information only (informative).

Informative sections (such as Rationale, Change History, Application Usage, and so on) are denoted by continuous shading bars in the margins.

Ranges of values are indicated with parentheses or brackets as follows:

- (a,b) means the range of all values from a to b, including neither a nor b.
- [a,b] means the range of all values from a to b, including a and b.
- [a,b) means the range of all values from a to b, including a, but not b.
- (a,b] means the range of all values from a to b, including b, but not a.

Participants

The Austin Group

This standard was prepared by the Austin Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society, The Open Group, and ISO/SC22 WG15. 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/SC22 WG15

Mark Brown, Organizational Representative, The Open Group

Cathy Hughes, Technical Editor

Austin Group Technical Reviewers

Peter Anvin

Bouazza Bachar

Theodore P. Baker

Walter Briscoe

Mark Brown

Dave Butenhof

Geoff Clare

Donald W. Cragun

Lee Damico

Ulrich Drepper

Paul Eggert

Joanna Farley

Clive D.W. Feather

Andrew Gollan

Michael Gonzalez

Joseph M. Gwinn

Jon Hitchcock

Yvette Ho Sang

Cathy Hughes

Lowell G. Johnson

Andrew Josey

Michael Kavanaugh

David Korn

Marc Aurele La France

Jim Meyering

Gary Miller

Finnbarr P. Murphy

Joseph S. Myers

Sandra O'Donnell

Frank Prindle

Curtis Royster Jr.

Glen Seeds

Keld Jorn Simonsen

Raja Srinivasan

Nicholas Stoughton

Donn S. Terry

Fred Tydeman

Peter Van Der Veen

James Youngman

Jim Zepeda

Jason Zions

Austin Group Working Group Members

Harold C. Adams	Michael Gonzalez	Sandra O'Donnell
Peter Anvin	Karen D. Gordon	Frank Prindle
Pierre-Jean Arcos	Joseph M. Gwinn	Francois Riche
Jay Ashford	Steven A. Haaser	John D. Riley
Bouazza Bachar	Charles E. Hammons	Andrew K. Roach
Theodore P. Baker	Chris J. Harding	Helmut Roth
Robert Barned	Barry Hedquist	Jaideep Roy
Joel Berman	Vincent E. Henley	Curtis Royster Jr.
David J. Blackwood	Karl Heubaum	Stephen C. Schwarm
Shirley Bockstahler-Brandt	Jon Hitchcock	Glen Seeds
James Bottomley	Yvette Ho Sang	Richard Seibel
Walter Briscoe	Niklas Holsti	David L. Shroads Jr.
Andries Brouwer	Thomas Hosmer	W. Olin Sibert
Mark Brown	Cathy Hughes	Keld Jorn Simonsen
Eric W. Burger	Jim D. Isaak	Curtis Smith
Alan Burns	Lowell G. Johnson	Raja Srinivasan
Andries Brouwer	Michael B. Jones	Nicholas Stoughton
Dave Butenhof	Andrew Josey	Marc J. Teller
Keith Chow	Michael J. Karels	Donn S. Terry
Geoff Clare	Michael Kavanaugh	Fred Tydeman
Donald W. Cragun	David Korn	Mark-Rene Uchida
Lee Damico	Steven Kramer	Scott A. Valcourt
Juan Antonio De La Puente	Thomas M. Kurihara	Peter Van Der Veen
Ming De Zhou	Marc Aurele La France	Michael W. Vannier
Steven J. Dovich	C. Douglass Locke	Eric Vought
Richard P. Draves	Nick MacLaren	Frederick N. Webb
Ulrich Drepper	Roger J. Martin	Paul A.T. Wolfgang
Paul Eggert	Craig H. Meyer	Garrett Wollman
Philip H. Enslow	Jim Meyering	James Youngman
Joanna Farley	Gary Miller	Oren Yuen
Clive D.W. Feather	Finnbarr P. Murphy	Janusz Zalewski
Pete Forman	Joseph S. Myers	Jim Zepeda
Mark Funkenhauser	John Napier	Jason Zions
Lois Goldthwaite	Peter E. Obermayer	
Andrew Gollan	James T. Oblinger	

Participants

The Open Group

When The Open Group approved the Base Specifications, Issue 6 on 12 September 2001, the membership of The Open Group Base Working Group was as follows.

Andrew Josey, Chair

Finnbarr P. Murphy, Vice-Chair

Mark Brown, Austin Group Liaison

Cathy Hughes, Technical Editor

Base Working Group Members

Bouazza Bachar

Mark Brown

Dave Butenhof

Donald W. Cragun

Larry Dwyer

Joanna Farley

Andrew Gollan

Karen D. Gordon

Gary Miller

Finnbarr P. Murphy

Frank Prindle

Andrew K. Roach

Curtis Royster Jr.

Nicholas Stoughton

Kenjiro Tsuji

IEEE

When the IEEE Standards Board approved IEEE Std 1003.1-2001 on 6 December 2001, 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. Balloters may have voted for approval, disapproval, or abstention.

Harold C. Adams	Steven A. Haaser	Frank Prindle
Pierre-Jean Arcos	Charles E. Hammons	Francois Riche
Jay Ashford	Chris J. Harding	John D. Riley
Theodore P. Baker	Barry Hedquist	Andrew K. Roach
Robert Barned	Vincent E. Henley	Helmut Roth
David J. Blackwood	Karl Heubaum	Jaideep Roy
Shirley Bockstahler-Brandt	Niklas Holsti	Curtis Royster Jr.
James Bottomley	Thomas Hosmer	Stephen C. Schwarm
Mark Brown	Jim D. Isaak	Richard Seibel
Eric W. Burger	Lowell G. Johnson	David L. Shroads Jr.
Alan Burns	Michael B. Jones	W. Olin Sibert
Dave Butenhof	Andrew Josey	Keld Jorn Simonsen
Keith Chow	Michael J. Karels	Nicholas Stoughton
Donald W. Cragun	Steven Kramer	Donn S. Terry
Juan Antonio De La Puente	Thomas M. Kurihara	Mark-Rene Uchida
Ming De Zhou	C. Douglass Locke	Scott A. Valcourt
Steven J. Dovich	Roger J. Martin	Michael W. Vannier
Richard P. Draves	Craig H. Meyer	Frederick N. Webb
Philip H. Enslow	Finnbarr P. Murphy	Paul A.T. Wolfgang
Michael Gonzalez	John Napier	Oren Yuen
Karen D. Gordon	Peter E. Obermayer	Janusz Zalewski
Joseph M. Gwinn	James T. Oblinger	

The following organizational representative voted on this standard:

Andrew Josey, X/Open Company Ltd.

Participants

IEEE-SA Standards Board

When the IEEE-SA Standards Board approved IEEE Std 1003.1-2001 on 6 December 2001, it had the following membership:

Donald N. Heirman, Chair
James T. Carlo, Vice-Chair
Judith Gorman, Secretary

Satish K. Aggarwal
Mark D. Bowman
Gary R. Engmann
Harold E. Epstein
H. Landis Floyd
Jay Forster*
Howard M. Frazier
Ruben D. Garzon

James H. Gurney
Richard J. Holleman
Lowell G. Johnson
Robert J. Kennelly
Joseph L. Koepfinger*
Peter H. Lips
L. Bruce McClung
Daleep C. Mohla

James W. Moore
Robert F. Munzner
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Donald W. Zipse

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

Alan Cookson, NIST Representative
Donald R. Volzka, TAB Representative
Yvette Ho Sang, Don Messina, Savoula Amanatidis, IEEE Project Editors

* Member Emeritus

Trademarks

The following information is given for the convenience of users of this standard and does not constitute endorsement of these products by The Open Group or the IEEE. There may be other products mentioned in the text that might be covered by trademark protection and readers are advised to verify them independently.

1003.1TM is a trademark of the Institute of Electrical and Electronic Engineers, Inc.

AIX[®] is a registered trademark of IBM Corporation.

AT&T[®] is a registered trademark of AT&T in the U.S.A. and other countries.

BSDTM is a trademark of the University of California, Berkeley, U.S.A.

Hewlett-Packard[®], HP[®], and HP-UX[®] are registered trademarks of Hewlett-Packard Company.

IBM[®] is a registered trademark of International Business Machines Corporation.

Motif[®], OSF/1[®], UNIX[®], and the “X Device” are registered trademarks and IT DialToneTM and The Open GroupTM are trademarks of The Open Group in the U.S. and other countries.

POSIX[®] is a registered trademark of the Institute of Electrical and Electronic Engineers, Inc.

Sun[®] and Sun Microsystems[®] are registered trademarks of Sun Microsystems, Inc.

/usr/group[®] is a registered trademark of UniForum, the International Network of UNIX System Users.

Acknowledgements

The contributions of the following organizations to the development of IEEE Std 1003.1-2001 are gratefully acknowledged:

- AT&T for permission to reproduce portions of its copyrighted System V Interface Definition (SVID) and material from the UNIX System V Release 2.0 documentation.
- The SC22 WG14 Committees.

This standard was prepared by the Austin Group, a joint working group of the IEEE, The Open Group, and ISO SC22 WG15.

Referenced Documents

Normative References

Normative references for this standard are defined in the Base Definitions volume.

Informative References

The following documents are referenced in this standard:

1984 /usr/group Standard

/usr/group Standards Committee, Santa Clara, CA, UniForum 1984.

Almasi and Gottlieb

George S. Almasi and Allan Gottlieb, *Highly Parallel Computing*, The Benjamin/Cummings Publishing Company, Inc., 1989, ISBN: 0-8053-0177-1.

ANSI C

American National Standard for Information Systems: Standard X3.159-1989, Programming Language C.

ANSI X3.226-1994

American National Standard for Information Systems: Standard X3.226-1994, Programming Language Common LISP.

Brawer

Steven Brawer, *Introduction to Parallel Programming*, Academic Press, 1989, ISBN: 0-12-128470-0.

DeRemer and Pennello Article

DeRemer, Frank and Pennello, Thomas J., *Efficient Computation of LALR(1) Look-Ahead Sets*, SigPlan Notices, Volume 15, No. 8, August 1979.

Draft ANSI X3J11.1

IEEE Floating Point draft report of ANSI X3J11.1 (NCEG).

FIPS 151-1

Federal Information Procurement Standard (FIPS) 151-1. Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language].

FIPS 151-2

Federal Information Procurement Standards (FIPS) 151-2, Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language].

HP-UX Manual

Hewlett-Packard HP-UX Release 9.0 Reference Manual, Third Edition, August 1992.

IEC 60559: 1989

IEC 60559: 1989, Binary Floating-Point Arithmetic for Microprocessor Systems (previously designated IEC 559: 1989).

IEEE Std 754-1985

IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic.

IEEE Std 854-1987

IEEE Std 854-1987, IEEE Standard for Radix-Independent Floating-Point Arithmetic.

Referenced Documents

- IEEE Std 1003.9-1992**
IEEE Std 1003.9-1992, IEEE Standard for Information Technology — POSIX FORTRAN 77 Language Interfaces — Part 1: Binding for System Application Program Interface API.
- IETF RFC 791**
Internet Protocol, Version 4 (IPv4), September 1981.
- IETF RFC 819**
The Domain Naming Convention for Internet User Applications, Z. Su, J. Postel, August 1982.
- IETF RFC 822**
Standard for the Format of ARPA Internet Text Messages, D.H. Crocker, August 1982.
- IETF RFC 919**
Broadcasting Internet Datagrams, J. Mogul, October 1984.
- IETF RFC 920**
Domain Requirements, J. Postel, J. Reynolds, October 1984.
- IETF RFC 921**
Domain Name System Implementation Schedule, J. Postel, October 1984.
- IETF RFC 922**
Broadcasting Internet Datagrams in the Presence of Subnets, J. Mogul, October 1984.
- IETF RFC 1034**
Domain Names — Concepts and Facilities, P. Mockapetris, November 1987.
- IETF RFC 1035**
Domain Names — Implementation and Specification, P. Mockapetris, November 1987.
- IETF RFC 1123**
Requirements for Internet Hosts — Application and Support, R. Braden, October 1989.
- IETF RFC 1886**
DNS Extensions to Support Internet Protocol, Version 6 (IPv6), C. Huitema, S. Thomson, December 1995.
- IETF RFC 2045**
Multipurpose Internet Mail Extensions (MIME), Part 1: Format of Internet Message Bodies, N. Freed, N. Borenstein, November 1996.
- IETF RFC 2373**
Internet Protocol, Version 6 (IPv6) Addressing Architecture, S. Deering, R. Hinden, July 1998.
- IETF RFC 2460**
Internet Protocol, Version 6 (IPv6), S. Deering, R. Hinden, December 1998.
- Internationalisation Guide**
Guide, July 1993, Internationalisation Guide, Version 2 (ISBN: 1-859120-02-4, G304), published by The Open Group.
- ISO C (1990)**
ISO/IEC 9899:1990, Programming Languages — C, including Amendment 1:1995 (E), C Integrity (Multibyte Support Extensions (MSE) for ISO C).
- ISO 2375:1985**
ISO 2375:1985, Data Processing — Procedure for Registration of Escape Sequences.

ISO 8652:1987

ISO 8652:1987, Programming Languages — Ada (technically identical to ANSI standard 1815A-1983).

ISO/IEC 1539:1990

ISO/IEC 1539:1990, Information Technology — Programming Languages — Fortran (technically identical to the ANSI X3.9-1978 standard [FORTRAN 77]).

ISO/IEC 4873:1991

ISO/IEC 4873:1991, Information Technology — ISO 8-bit Code for Information Interchange — Structure and Rules for Implementation.

ISO/IEC 6429:1992

ISO/IEC 6429:1992, Information Technology — Control Functions for Coded Character Sets.

ISO/IEC 6937:1994

ISO/IEC 6937:1994, Information Technology — Coded Character Set for Text Communication — Latin Alphabet.

ISO/IEC 8802-3:1996

ISO/IEC 8802-3:1996, Information Technology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

ISO/IEC 8859

ISO/IEC 8859, Information Technology — 8-Bit Single-Byte Coded Graphic Character Sets:

- Part 1: Latin Alphabet No. 1
- Part 2: Latin Alphabet No. 2
- Part 3: Latin Alphabet No. 3
- Part 4: Latin Alphabet No. 4
- Part 5: Latin/Cyrillic Alphabet
- Part 6: Latin/Arabic Alphabet
- Part 7: Latin/Greek Alphabet
- Part 8: Latin/Hebrew Alphabet
- Part 9: Latin Alphabet No. 5
- Part 10: Latin Alphabet No. 6
- Part 13: Latin Alphabet No. 7
- Part 14: Latin Alphabet No. 8
- Part 15: Latin Alphabet No. 9

ISO POSIX-1:1996

ISO/IEC 9945-1:1996, Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language] (identical to ANSI/IEEE Std 1003.1-1996). Incorporating ANSI/IEEE Stds 1003.1-1990, 1003.1b-1993, 1003.1c-1995, and 1003.1i-1995.

ISO POSIX-2:1993

ISO/IEC 9945-2:1993, Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities (identical to ANSI/IEEE Std 1003.2-1992, as amended by ANSI/IEEE Std 1003.2a-1992).

Issue 1

X/Open Portability Guide, July 1985 (ISBN: 0-444-87839-4).

Referenced Documents

Issue 2

X/Open Portability Guide, January 1987:

- Volume 1: XVS Commands and Utilities (ISBN: 0-444-70174-5)
- Volume 2: XVS System Calls and Libraries (ISBN: 0-444-70175-3)

Issue 3

X/Open Specification, 1988, 1989, February 1992:

- Commands and Utilities, Issue 3 (ISBN: 1-872630-36-7, C211); this specification was formerly X/Open Portability Guide, Issue 3, Volume 1, January 1989, XSI Commands and Utilities (ISBN: 0-13-685835-X, XO/XPG/89/002)
- System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this specification was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003)
- Curses Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapters 9 to 14 inclusive; this specification was formerly X/Open Portability Guide, Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)
- Headers Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapter 19, Cpio and Tar Headers; this specification was formerly X/Open Portability Guide Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)

Issue 4

CAE Specification, July 1992, published by The Open Group:

- System Interface Definitions (XBD), Issue 4 (ISBN: 1-872630-46-4, C204)
- Commands and Utilities (XCU), Issue 4 (ISBN: 1-872630-48-0, C203)
- System Interfaces and Headers (XSH), Issue 4 (ISBN: 1-872630-47-2, C202)

Issue 4, Version 2

CAE Specification, August 1994, published by The Open Group:

- System Interface Definitions (XBD), Issue 4, Version 2 (ISBN: 1-85912-036-9, C434)
- Commands and Utilities (XCU), Issue 4, Version 2 (ISBN: 1-85912-034-2, C436)
- System Interfaces and Headers (XSH), Issue 4, Version 2 (ISBN: 1-85912-037-7, C435)

Issue 5

Technical Standard, February 1997, published by The Open Group:

- System Interface Definitions (XBD), Issue 5 (ISBN: 1-85912-186-1, C605)
- Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)
- System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)

Knuth Article

Knuth, Donald E., *On the Translation of Languages from Left to Right*, Information and Control, Volume 8, No. 6, October 1965.

KornShell

Bolsky, Morris I. and Korn, David G., *The New KornShell Command and Programming Language*, March 1995, Prentice Hall.

MSE Working Draft

Working draft of ISO/IEC 9899:1990/Add3: Draft, Addendum 3 — Multibyte Support Extensions (MSE) as documented in the ISO Working Paper SC22/WG14/N205 dated 31 March 1992.

POSIX.0: 1995

IEEE Std 1003.0-1995, IEEE Guide to the POSIX Open System Environment (OSE) (identical to ISO/IEC TR 14252).

POSIX.1: 1988

IEEE Std 1003.1-1988, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].

POSIX.1: 1990

IEEE Std 1003.1-1990, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].

POSIX.1a

P1003.1a, Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — (C Language) Amendment

POSIX.1d: 1999

IEEE Std 1003.1d-1999, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 4: Additional Realtime Extensions [C Language].

POSIX.1g: 2000

IEEE Std 1003.1g-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 6: Protocol-Independent Interfaces (PII).

POSIX.1j: 2000

IEEE Std 1003.1j-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 5: Advanced Realtime Extensions [C Language].

POSIX.1q: 2000

IEEE Std 1003.1q-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 7: Tracing [C Language].

POSIX.2b

P1003.2b, Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities — Amendment

POSIX.2d: 1994

IEEE Std 1003.2d: 1994, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities — Amendment 1: Batch Environment.

POSIX.13: 1998

IEEE Std 1003.13: 1998, IEEE Standard for Information Technology — Standardized Application Environment Profile (AEP) — POSIX Realtime Application Support.

Referenced Documents

Sarwate Article

Sarwate, Dilip V., *Computation of Cyclic Redundancy Checks via Table Lookup*, Communications of the ACM, Volume 30, No. 8, August 1988.

Sprung, Sha, and Lehoczky

Sprung, B., Sha, L., and Lehoczky, J.P., *Aperiodic Task Scheduling for Hard Real-Time Systems*, The Journal of Real-Time Systems, Volume 1, 1989, Pages 27-60.

SVID, Issue 1

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 1; Morristown, NJ, UNIX Press, 1985.

SVID, Issue 2

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 2; Morristown, NJ, UNIX Press, 1986.

SVID, Issue 3

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 3; Morristown, NJ, UNIX Press, 1989.

The AWK Programming Language

Aho, Alfred V., Kernighan, Brian W., and Weinberger, Peter J., *The AWK Programming Language*, Reading, MA, Addison-Wesley 1988.

UNIX Programmer's Manual

American Telephone and Telegraph Company, *UNIX Time-Sharing System: UNIX Programmer's Manual*, 7th Edition, Murray Hill, NJ, Bell Telephone Laboratories, January 1979.

XNS, Issue 4

CAE Specification, August 1994, Networking Services, Issue 4 (ISBN: 1-85912-049-0, C438), published by The Open Group.

XNS, Issue 5

CAE Specification, February 1997, Networking Services, Issue 5 (ISBN: 1-85912-165-9, C523), published by The Open Group.

XNS, Issue 5.2

Technical Standard, January 2000, Networking Services (XNS), Issue 5.2 (ISBN: 1-85912-241-8, C808), published by The Open Group.

X/Open Courses, Issue 4, Version 2

CAE Specification, May 1996, X/Open Courses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), published by The Open Group.

Yacc

Yacc: Yet Another Compiler Compiler, Stephen C. Johnson, 1978.

Source Documents

Parts of the following documents were used to create the base documents for this standard:

AIX 3.2 Manual

AIX Version 3.2 For RISC System/6000, Technical Reference: Base Operating System and Extensions, 1990, 1992 (Part No. SC23-2382-00).

OSF/1

OSF/1 Programmer's Reference, Release 1.2 (ISBN: 0-13-020579-6).

OSF AES

Application Environment Specification (AES) Operating System Programming Interfaces Volume, Revision A (ISBN: 0-13-043522-8).

System V Release 2.0

- UNIX System V Release 2.0 Programmer's Reference Manual (April 1984 - Issue 2).
- UNIX System V Release 2.0 Programming Guide (April 1984 - Issue 2).

System V Release 4.2

Operating System API Reference, UNIX SVR4.2 (1992) (ISBN: 0-13-017658-3).

Introduction

1.1 Scope

2 The scope of IEEE Std 1003.1-2001 is described in the Base Definitions volume of
3 IEEE Std 1003.1-2001.
4

5 1.2 Conformance

6 Conformance requirements for IEEE Std 1003.1-2001 are defined in the Base Definitions volume
7 of IEEE Std 1003.1-2001, Chapter 2, Conformance.

8 1.3 Normative References

9 Normative references for IEEE Std 1003.1-2001 are defined in the Base Definitions volume of
10 IEEE Std 1003.1-2001.

11 1.4 Change History

12 Change history is described in the Rationale (Informative) volume of IEEE Std 1003.1-2001, and
13 in the CHANGE HISTORY section of reference pages.

14 1.5 Terminology

15 This section appears in the Base Definitions volume of IEEE Std 1003.1-2001, but is repeated here
16 for convenience:

17 For the purposes of IEEE Std 1003.1-2001, the following terminology definitions apply:

18 **can**

19 Describes a permissible optional feature or behavior available to the user or application. The
20 feature or behavior is mandatory for an implementation that conforms to
21 IEEE Std 1003.1-2001. An application can rely on the existence of the feature or behavior.

22 **implementation-defined**

23 Describes a value or behavior that is not defined by IEEE Std 1003.1-2001 but is selected by
24 an implementor. The value or behavior may vary among implementations that conform to
25 IEEE Std 1003.1-2001. An application should not rely on the existence of the value or
26 behavior. An application that relies on such a value or behavior cannot be assured to be
27 portable across conforming implementations.

28 The implementor shall document such a value or behavior so that it can be used correctly
29 by an application.

30 **legacy**

31 Describes a feature or behavior that is being retained for compatibility with older
32 applications, but which has limitations which make it inappropriate for developing portable

33 applications. New applications should use alternative means of obtaining equivalent
34 functionality.

35 **may**
36 Describes a feature or behavior that is optional for an implementation that conforms to
37 IEEE Std 1003.1-2001. An application should not rely on the existence of the feature or
38 behavior. An application that relies on such a feature or behavior cannot be assured to be
39 portable across conforming implementations.

40 To avoid ambiguity, the opposite of *may* is expressed as *need not*, instead of *may not*.

41 **shall**
42 For an implementation that conforms to IEEE Std 1003.1-2001, describes a feature or
43 behavior that is mandatory. An application can rely on the existence of the feature or
44 behavior.

45 For an application or user, describes a behavior that is mandatory.

46 **should**
47 For an implementation that conforms to IEEE Std 1003.1-2001, describes a feature or
48 behavior that is recommended but not mandatory. An application should not rely on the
49 existence of the feature or behavior. An application that relies on such a feature or behavior
50 cannot be assured to be portable across conforming implementations.

51 For an application, describes a feature or behavior that is recommended programming
52 practice for optimum portability.

53 **undefined**
54 Describes the nature of a value or behavior not defined by IEEE Std 1003.1-2001 which
55 results from use of an invalid program construct or invalid data input.

56 The value or behavior may vary among implementations that conform to
57 IEEE Std 1003.1-2001. An application should not rely on the existence or validity of the
58 value or behavior. An application that relies on any particular value or behavior cannot be
59 assured to be portable across conforming implementations.

60 **unspecified**
61 Describes the nature of a value or behavior not specified by IEEE Std 1003.1-2001 which
62 results from use of a valid program construct or valid data input.

63 The value or behavior may vary among implementations that conform to
64 IEEE Std 1003.1-2001. An application should not rely on the existence or validity of the
65 value or behavior. An application that relies on any particular value or behavior cannot be
66 assured to be portable across conforming implementations.

67 **1.6 Definitions**

68 Concepts and definitions are defined in the Base Definitions volume of IEEE Std 1003.1-2001.

69 **1.7 Relationship to Other Documents**

70 **1.7.1 System Interfaces**

71 This subsection describes some of the features provided by the System Interfaces volume of
72 IEEE Std 1003.1-2001 that are assumed to be globally available on all systems conforming to this
73 volume of IEEE Std 1003.1-2001. This subsection does not attempt to detail all of the features
74 defined in the System Interfaces volume of IEEE Std 1003.1-2001 that are required by all of the
75 utilities defined in this volume of IEEE Std 1003.1-2001; the utility and function descriptions
76 point out additional functionality required to provide the corresponding specific features
77 needed by each.

78 The following subsections describe frequently used concepts. Many of these concepts are
79 described in the Base Definitions volume of IEEE Std 1003.1-2001. Utility and function
80 description statements override these defaults when appropriate.

81 **1.7.1.1 Process Attributes**

82 The following process attributes, as described in the System Interfaces volume of
83 IEEE Std 1003.1-2001, are assumed to be supported for all processes in this volume of
84 IEEE Std 1003.1-2001:

Controlling Terminal	Real Group ID
Current Working Directory	Real User ID
Effective Group ID	Root Directory
Effective User ID	Saved Set-Group-ID
File Descriptors	Saved Set-User-ID
File Mode Creation Mask	Session Membership
Process Group ID	Supplementary Group IDs
Process ID	

93 A conforming implementation may include additional process attributes.

94 **1.7.1.2 Concurrent Execution of Processes**

95 The following functionality of the *fork()* function defined in the System Interfaces volume of
96 IEEE Std 1003.1-2001 shall be available on all systems conforming to this volume of
97 IEEE Std 1003.1-2001:

- 98 1. Independent processes shall be capable of executing independently without either process
99 terminating.
- 100 2. A process shall be able to create a new process with all of the attributes referenced in
101 Section 1.7.1.1, determined according to the semantics of a call to the *fork()* function
102 defined in the System Interfaces volume of IEEE Std 1003.1-2001 followed by a call in the
103 child process to one of the *exec* functions defined in the System Interfaces volume of
104 IEEE Std 1003.1-2001.

105 1.7.1.3 File Access Permissions

106 The file access control mechanism described by the Base Definitions volume of
107 IEEE Std 1003.1-2001, Section 4.4, File Access Permissions shall apply to all files on an
108 implementation conforming to this volume of IEEE Std 1003.1-2001.

109 1.7.1.4 File Read, Write, and Creation

110 If a file that does not exist is to be written, it shall be created as described below, unless the
111 utility description states otherwise.

112 When a file that does not exist is created, the following features defined in the System Interfaces
113 volume of IEEE Std 1003.1-2001 shall apply unless the utility or function description states
114 otherwise:

- 115 1. The user ID of the file shall be set to the effective user ID of the calling process.
- 116 2. The group ID of the file shall be set to the effective group ID of the calling process or the
117 group ID of the directory in which the file is being created.
- 118 3. If the file is a regular file, the permission bits of the file shall be set to:

119 S_IROTH | S_IWOTH | S_IRGRP | S_IWGRP | S_IRUSR | S_IWUSR

120 (see the description of *File Modes* in the Base Definitions volume of IEEE Std 1003.1-2001,
121 Chapter 13, Headers, <sys/stat.h>) except that the bits specified by the file mode creation
122 mask of the process shall be cleared. If the file is a directory, the permission bits shall be set
123 to:

124 S_IRWXU | S_IRWXG | S_IRWXO

125 except that the bits specified by the file mode creation mask of the process shall be cleared.

- 126 4. The *st_atime*, *st_ctime*, and *st_mtime* fields of the file shall be updated as specified in the
127 System Interfaces volume of IEEE Std 1003.1-2001, Section 2.5, Standard I/O Streams.
- 128 5. If the file is a directory, it shall be an empty directory; otherwise, the file shall have length
129 zero.
- 130 6. If the file is a symbolic link, the effect shall be undefined unless the {POSIX2_SYMLINKS}
131 variable is in effect for the directory in which the symbolic link would be created.
- 132 7. Unless otherwise specified, the file created shall be a regular file.

133 When an attempt is made to create a file that already exists, the utility shall take the action
134 indicated in Table 1-1 (on page 5) corresponding to the type of the file the utility is trying to
135 create and the type of the existing file, unless the utility description states otherwise.

136

Table 1-1 Actions when Creating a File that Already Exists

Existing Type	New Type										Function Creating New
	B	C	D	F	L	M	P	Q	R	S	
A <i>fattach()</i> -ed STREAM	F	F	F	F	F	—	—	OF	—	—	N/A
B Block Special	F	F	F	F	F	—	—	OF	—	—	<i>mknod()</i> **
C Character Special	F	F	F	F	F	—	—	OF	—	—	<i>mknod()</i> **
D Directory	F	F	F	F	F	—	—	F	—	—	<i>mkdir()</i>
F FIFO Special File	F	F	F	F	F	—	—	O	—	—	<i>mkfifo()</i>
L Symbolic Link	F	F	F	F	F	—	—	FL	—	—	<i>symlink()</i>
M Shared Memory	F	F	F	F	F	—	—	—	—	—	<i>shm_open()</i>
P Semaphore	F	F	F	F	F	—	—	—	—	—	<i>sem_open()</i>
Q Message Queue	F	F	F	F	F	—	—	—	—	—	<i>mq_open()</i>
R Regular File	F	F	F	F	F	—	—	RF	—	—	<i>open()</i>
S Socket	F	F	F	F	F	—	—	—	—	—	<i>bind()</i>
T Typed Memory	F	F	F	F	F	—	—	—	—	—	*

151

The following codes are used in Table 1-1:

152

F Fail. The attempt to create the new file shall fail and the utility shall either continue with its operation or exit immediately with a non-zero exit status, depending on the description of the utility.

155

FL Follow link. Unless otherwise specified, the symbolic link shall be followed as specified for pathname resolution, and the operation performed shall be as if the target of the symbolic link (after all resolution) had been named. If the target of the symbolic link does not exist, it shall be as if that nonexistent target had been named directly.

159

O Open FIFO. When attempting to create a regular file, and the existing file is a FIFO special file:

161

1. If the FIFO is not already open for reading, the attempt shall block until the FIFO is opened for reading.

163

2. Once the FIFO is open for reading, the utility shall open the FIFO for writing and continue with its operation.

165

OF The named file shall be opened with the consequences defined for that file type.

166

RF Regular file. When attempting to create a regular file, and the existing file is a regular file:

167

1. The user ID, group ID, and permission bits of the file shall not be changed.

168

2. The file shall be truncated to zero length.

169

3. The *st_ctime* and *st_mtime* fields shall be marked for update.

170

— The effect is implementation-defined unless specified by the utility description.

171

* There is no portable way to create a file of this type.

172

** Not portable.

173

When a file is to be appended, the file shall be opened in a manner equivalent to using the O_APPEND flag, without the O_TRUNC flag, in the *open()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001.

176

When a file is to be read or written, the file shall be opened with an access mode corresponding to the operation to be performed. If file access permissions deny access, the requested operation shall fail.

179 1.7.1.5 *File Removal*

180 When a directory that is the root directory or current working directory of any process is
181 removed, the effect is implementation-defined. If file access permissions deny access, the
182 requested operation shall fail. Otherwise, when a file is removed:

- 183 1. Its directory entry shall be removed from the file system.
- 184 2. The link count of the file shall be decremented.
- 185 3. If the file is an empty directory (see the Base Definitions volume of IEEE Std 1003.1-2001,
186 Section 3.143, Empty Directory):
 - 187 a. If no process has the directory open, the space occupied by the directory shall be
188 freed and the directory shall no longer be accessible.
 - 189 b. If one or more processes have the directory open, the directory contents shall be
190 preserved until all references to the file have been closed.
- 191 4. If the file is a directory that is not empty, the *st_ctime* field shall be marked for update.
- 192 5. If the file is not a directory:
 - 193 a. If the link count becomes zero:
 - 194 i. If no process has the file open, the space occupied by the file shall be freed and
195 the file shall no longer be accessible.
 - 196 ii. If one or more processes have the file open, the file contents shall be preserved
197 until all references to the file have been closed.
 - 198 b. If the link count is not reduced to zero, the *st_ctime* field shall be marked for update.
- 199 6. The *st_ctime* and *st_mtime* fields of the containing directory shall be marked for update.

200 1.7.1.6 *File Time Values*

201 All files shall have the three time values described by the Base Definitions volume of
202 IEEE Std 1003.1-2001, Section 4.7, File Times Update.

203 1.7.1.7 *File Contents*

204 When a reference is made to the contents of a file, *pathname*, this means the equivalent of all of
205 the data placed in the space pointed to by *buf* when performing the *read()* function calls in the
206 following operations defined in the System Interfaces volume of IEEE Std 1003.1-2001:

```
207     while (read (fildes, buf, nbytes) > 0)
208         ;
```

209 If the file is indicated by a pathname *pathname*, the file descriptor shall be determined by the
210 equivalent of the following operation defined in the System Interfaces volume of
211 IEEE Std 1003.1-2001:

```
212     fildes = open (pathname, O_RDONLY);
```

213 The value of *nbytes* in the above sequence is unspecified; if the file is of a type where the data
214 returned by *read()* would vary with different values, the value shall be one that results in the
215 most data being returned.

216 If the *read()* function calls would return an error, it is unspecified whether the contents of the file
217 are considered to include any data from offsets in the file beyond where the error would be
218 returned.

219 1.7.1.8 *Pathname Resolution*

220 The pathname resolution algorithm, described by the Base Definitions volume of
221 IEEE Std 1003.1-2001, Section 4.11, Pathname Resolution, shall be used by implementations
222 conforming to this volume of IEEE Std 1003.1-2001; see also the Base Definitions volume of
223 IEEE Std 1003.1-2001, Section 4.5, File Hierarchy.

224 1.7.1.9 *Changing the Current Working Directory*

225 When the current working directory (see the Base Definitions volume of IEEE Std 1003.1-2001,
226 Section 3.436, Working Directory) is to be changed, unless the utility or function description
227 states otherwise, the operation shall succeed unless a call to the *chdir()* function defined in the
228 System Interfaces volume of IEEE Std 1003.1-2001 would fail when invoked with the new
229 working directory pathname as its argument.

230 1.7.1.10 *Establish the Locale*

231 The functionality of the *setlocale()* function defined in the System Interfaces volume of
232 IEEE Std 1003.1-2001 shall be available on all systems conforming to this volume of
233 IEEE Std 1003.1-2001; that is, utilities that require the capability of establishing an international
234 operating environment shall be permitted to set the specified category of the international
235 environment.

236 1.7.1.11 *Actions Equivalent to Functions*

237 Some utility descriptions specify that a utility performs actions equivalent to a function defined
238 in the System Interfaces volume of IEEE Std 1003.1-2001. Such specifications require only that
239 the external effects be equivalent, not that any effect within the utility and visible only to the
240 utility be equivalent.

241 1.7.2 **Concepts Derived from the ISO C Standard**

242 Some of the standard utilities perform complex data manipulation using their own procedure
243 and arithmetic languages, as defined in their EXTENDED DESCRIPTION or OPERANDS
244 sections. Unless otherwise noted, the arithmetic and semantic concepts (precision, type
245 conversion, control flow, and so on) shall be equivalent to those defined in the ISO C standard,
246 as described in the following sections. Note that there is no requirement that the standard
247 utilities be implemented in any particular programming language.

248 1.7.2.1 *Arithmetic Precision and Operations*

249 Integer variables and constants, including the values of operands and option-arguments, used
250 by the standard utilities listed in this volume of IEEE Std 1003.1-2001 shall be implemented as
251 equivalent to the ISO C standard **signed long** data type; floating point shall be implemented as
252 equivalent to the ISO C standard **double** type. Conversions between types shall be as described
253 in the ISO C standard. All variables shall be initialized to zero if they are not otherwise assigned
254 by the input to the application.

255 Arithmetic operators and control flow keywords shall be implemented as equivalent to those in
256 the cited ISO C standard section, as listed in Table 1-2 (on page 8).

Table 1-2 Selected ISO C Standard Operators and Control Flow Keywords

Operation	ISO C Standard Equivalent Reference
<code>()</code>	Section 6.5.1, Primary Expressions
<code>postfix ++</code> <code>postfix --</code>	Section 6.5.2, Postfix Operators
<code>unary +</code> <code>unary -</code> <code>prefix ++</code> <code>prefix --</code> <code>~</code> <code>!</code> <code>sizeof()</code>	Section 6.5.3, Unary Operators
<code>*</code> <code>/</code> <code>%</code>	Section 6.5.5, Multiplicative Operators
<code>+</code> <code>-</code>	Section 6.5.6, Additive Operators
<code><<</code> <code>>></code>	Section 6.5.7, Bitwise Shift Operators
<code><, <=</code> <code>>, >=</code>	Section 6.5.8, Relational Operators
<code>==</code> <code>!=</code>	Section 6.5.9, Equality Operators
<code>&</code>	Section 6.5.10, Bitwise AND Operator
<code>^</code>	Section 6.5.11, Bitwise Exclusive OR Operator
<code> </code>	Section 6.5.12, Bitwise Inclusive OR Operator
<code>&&</code>	Section 6.5.13, Logical AND Operator
<code> </code>	Section 6.5.14, Logical OR Operator
<code>expr?expr:expr</code>	Section 6.5.15, Conditional Operator
<code>=, *=, /=, %=, +=, -=</code> <code><<=, >>=, &=, ^=, =</code>	Section 6.5.16, Assignment Operators
<code>if ()</code> <code>if () ... else</code> <code>switch ()</code>	Section 6.8.4, Selection Statements
<code>while ()</code> <code>do ... while ()</code> <code>for ()</code>	Section 6.8.5, Iteration Statements
<code>goto</code> <code>continue</code> <code>break</code> <code>return</code>	Section 6.8.6, Jump Statements

The evaluation of arithmetic expressions shall be equivalent to that described in Section 6.5, Expressions, of the ISO C standard.

301 **1.7.2.2 Mathematical Functions**

302 Any mathematical functions with the same names as those in the following sections of the ISO C
303 standard:

- 304 • Section 7.12, Mathematics, <math.h>
- 305 • Section 7.20.2, Pseudo-Random Sequence Generation Functions

306 shall be implemented to return the results equivalent to those returned from a call to the
307 corresponding function described in the ISO C standard.

308 **1.8 Portability**

309 Some of the utilities in the Shell and Utilities volume of IEEE Std 1003.1-2001 and functions in
310 the System Interfaces volume of IEEE Std 1003.1-2001 describe functionality that might not be
311 fully portable to systems meeting the requirements for POSIX conformance (see the Base
312 Definitions volume of IEEE Std 1003.1-2001, Chapter 2, Conformance).

313 Where optional, enhanced, or reduced functionality is specified, the text is shaded and a code in
314 the margin identifies the nature of the option, extension, or warning (see Section 1.8.1). For
315 maximum portability, an application should avoid such functionality.

316 Unless the primary task of a utility is to produce textual material on its standard output,
317 application developers should not rely on the format or content of any such material that may be
318 produced. Where the primary task *is* to provide such material, but the output format is
319 incompletely specified, the description is marked with the OF margin code and shading.
320 Application developers are warned not to expect that the output of such an interface on one
321 system is any guide to its behavior on another system.

322 **1.8.1 Codes**

323 Codes and their meanings are listed in the Base Definitions volume of IEEE Std 1003.1-2001, but
324 are repeated here for convenience:

325 ADV **Advisory Information**

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

328 Where applicable, functions are marked with the ADV margin legend in the SYNOPSIS section.
329 Where additional semantics apply to a function, the material is identified by use of the ADV
330 margin legend.

331 AIO **Asynchronous Input and Output**

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

334 Where applicable, functions are marked with the AIO margin legend in the SYNOPSIS section.
335 Where additional semantics apply to a function, the material is identified by use of the AIO
336 margin legend.

337 BAR **Barriers**

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

340 Where applicable, functions are marked with the BAR margin legend in the SYNOPSIS section.
341 Where additional semantics apply to a function, the material is identified by use of the BAR
342 margin legend.

343	BE	Batch Environment Services and Utilities
344		The functionality described is optional.
345		Where applicable, utilities are marked with the BE margin legend in the SYNOPSIS section.
346		Where additional semantics apply to a utility, the material is identified by use of the BE margin legend.
348	CD	C-Language Development Utilities
349		The functionality described is optional.
350		Where applicable, utilities are marked with the CD margin legend in the SYNOPSIS section.
351		Where additional semantics apply to a utility, the material is identified by use of the CD margin legend.
353	CPT	Process CPU-Time Clocks
354		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
356		Where applicable, functions are marked with the CPT margin legend in the SYNOPSIS section.
357		Where additional semantics apply to a function, the material is identified by use of the CPT margin legend.
359	CS	Clock Selection
360		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
362		Where applicable, functions are marked with the CS margin legend in the SYNOPSIS section.
363		Where additional semantics apply to a function, the material is identified by use of the CS margin legend.
365	CX	Extension to the ISO C standard
366		The functionality described is an extension to the ISO C standard. Application writers may make use of an extension as it is supported on all IEEE Std 1003.1-2001-conforming systems.
368		With each function or header from the ISO C standard, a statement to the effect that “any conflict is unintentional” is included. That is intended to refer to a direct conflict.
369		IEEE Std 1003.1-2001 acts in part as a profile of the ISO C standard, and it may choose to further constrain behaviors allowed to vary by the ISO C standard. Such limitations are not considered conflicts.
373		Where additional semantics apply to a function or header, the material is identified by use of the CX margin legend.
375	FD	FORTRAN Development Utilities
376		The functionality described is optional.
377		Where applicable, utilities are marked with the FD margin legend in the SYNOPSIS section.
378		Where additional semantics apply to a utility, the material is identified by use of the FD margin legend.
380	FR	FORTRAN Runtime Utilities
381		The functionality described is optional.
382		Where applicable, utilities are marked with the FR margin legend in the SYNOPSIS section.
383		Where additional semantics apply to a utility, the material is identified by use of the FR margin legend.
385	FSC	File Synchronization
386		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
387		

388	Where applicable, functions are marked with the FSC margin legend in the SYNOPSIS section.
389	Where additional semantics apply to a function, the material is identified by use of the FSC margin legend.
390	
391	IPV6
392	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
393	
394	Where applicable, functions are marked with the IP6 margin legend in the SYNOPSIS section.
395	Where additional semantics apply to a function, the material is identified by use of the IP6 margin legend.
396	
397	MC1 Advisory Information and either Memory Mapped Files or Shared Memory Objects
398	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
399	
400	This is a shorthand notation for combinations of multiple option codes.
401	Where applicable, functions are marked with the MC1 margin legend in the SYNOPSIS section.
402	Where additional semantics apply to a function, the material is identified by use of the MC1 margin legend.
403	
404	Refer to the Base Definitions volume of IEEE Std 1003.1-2001, Section 1.5.2, Margin Code Notation.
405	
406	MC2 Memory Mapped Files, Shared Memory Objects, or Memory Protection
407	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
408	
409	This is a shorthand notation for combinations of multiple option codes.
410	Where applicable, functions are marked with the MC2 margin legend in the SYNOPSIS section.
411	Where additional semantics apply to a function, the material is identified by use of the MC2 margin legend.
412	
413	Refer to the Base Definitions volume of IEEE Std 1003.1-2001, Section 1.5.2, Margin Code Notation.
414	
415	MF Memory Mapped Files
416	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
417	
418	Where applicable, functions are marked with the MF margin legend in the SYNOPSIS section.
419	Where additional semantics apply to a function, the material is identified by use of the MF margin legend.
420	
421	ML Process Memory Locking
422	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
423	
424	Where applicable, functions are marked with the ML margin legend in the SYNOPSIS section.
425	Where additional semantics apply to a function, the material is identified by use of the ML margin legend.
426	
427	MLR Range Memory Locking
428	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
429	
430	Where applicable, functions are marked with the MLR margin legend in the SYNOPSIS section.
431	Where additional semantics apply to a function, the material is identified by use of the MLR

432	margin legend.
433	MON Monotonic Clock The functionality described is optional. The functionality described is also an extension to the ISO C standard.
436	Where applicable, functions are marked with the MON margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the MON margin legend.
439	MPR Memory Protection The functionality described is optional. The functionality described is also an extension to the ISO C standard.
442	Where applicable, functions are marked with the MPR margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the MPR margin legend.
445	MSG Message Passing The functionality described is optional. The functionality described is also an extension to the ISO C standard.
448	Where applicable, functions are marked with the MSG margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the MSG margin legend.
451	MX IEC 60559 Floating-Point Option The functionality described is optional. The functionality described is also an extension to the ISO C standard.
454	Where applicable, functions are marked with the MX margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the MX margin legend.
457	OB Obsolescent The functionality described may be withdrawn in a future version of this volume of IEEE Std 1003.1-2001. Strictly Conforming POSIX Applications and Strictly Conforming XSI Applications shall not use obsolescent features.
461	Where applicable, the material is identified by use of the OB margin legend.
462	OF Output Format Incompletely Specified The functionality described is an XSI extension. The format of the output produced by the utility is not fully specified. It is therefore not possible to post-process this output in a consistent fashion. Typical problems include unknown length of strings and unspecified field delimiters.
466	Where applicable, the material is identified by use of the OF margin legend.
467	OH Optional Header In the SYNOPSIS section of some interfaces in the System Interfaces volume of IEEE Std 1003.1-2001 an included header is marked as in the following example:
470	#include <sys/types.h> #include <grp.h> struct group *getgrnam(const char *name);
473	The OH margin legend indicates that the marked header is not required on XSI-conformant systems.

475	PIO	Prioritized Input and Output
476		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
478		Where applicable, functions are marked with the PIO margin legend in the SYNOPSIS section.
479		Where additional semantics apply to a function, the material is identified by use of the PIO margin legend.
481	PS	Process Scheduling
482		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
484		Where applicable, functions are marked with the PS margin legend in the SYNOPSIS section.
485		Where additional semantics apply to a function, the material is identified by use of the PS margin legend.
487	RS	Raw Sockets
488		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
490		Where applicable, functions are marked with the RS margin legend in the SYNOPSIS section.
491		Where additional semantics apply to a function, the material is identified by use of the RS margin legend.
493	RTS	Realtime Signals Extension
494		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
496		Where applicable, functions are marked with the RTS margin legend in the SYNOPSIS section.
497		Where additional semantics apply to a function, the material is identified by use of the RTS margin legend.
499	SD	Software Development Utilities
500		The functionality described is optional.
501		Where applicable, utilities are marked with the SD margin legend in the SYNOPSIS section.
502		Where additional semantics apply to a utility, the material is identified by use of the SD margin legend.
504	SEM	Semaphores
505		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
507		Where applicable, functions are marked with the SEM margin legend in the SYNOPSIS section.
508		Where additional semantics apply to a function, the material is identified by use of the SEM margin legend.
510	SHM	Shared Memory Objects
511		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
513		Where applicable, functions are marked with the SHM margin legend in the SYNOPSIS section.
514		Where additional semantics apply to a function, the material is identified by use of the SHM margin legend.
516	SIO	Synchronized Input and Output
517		The functionality described is optional. The functionality described is also an extension to the ISO C standard.

519		Where applicable, functions are marked with the SIO margin legend in the SYNOPSIS section.
520		Where additional semantics apply to a function, the material is identified by use of the SIO margin legend.
522	SPI	Spin Locks The functionality described is optional. The functionality described is also an extension to the ISO C standard.
525		Where applicable, functions are marked with the SPI margin legend in the SYNOPSIS section.
526		Where additional semantics apply to a function, the material is identified by use of the SPI margin legend.
528	SPN	Spawn The functionality described is optional. The functionality described is also an extension to the ISO C standard.
531		Where applicable, functions are marked with the SPN margin legend in the SYNOPSIS section.
532		Where additional semantics apply to a function, the material is identified by use of the SPN margin legend.
534	SS	Process Sporadic Server The functionality described is optional. The functionality described is also an extension to the ISO C standard.
537		Where applicable, functions are marked with the SS margin legend in the SYNOPSIS section.
538		Where additional semantics apply to a function, the material is identified by use of the SS margin legend.
540	TCT	Thread CPU-Time Clocks The functionality described is optional. The functionality described is also an extension to the ISO C standard.
543		Where applicable, functions are marked with the TCT margin legend in the SYNOPSIS section.
544		Where additional semantics apply to a function, the material is identified by use of the TCT margin legend.
546	TEF	Trace Event Filter The functionality described is optional. The functionality described is also an extension to the ISO C standard.
549		Where applicable, functions are marked with the TEF margin legend in the SYNOPSIS section.
550		Where additional semantics apply to a function, the material is identified by use of the TEF margin legend.
552	THR	Threads The functionality described is optional. The functionality described is also an extension to the ISO C standard.
555		Where applicable, functions are marked with the THR margin legend in the SYNOPSIS section.
556		Where additional semantics apply to a function, the material is identified by use of the THR margin legend.
558	TMO	Timeouts The functionality described is optional. The functionality described is also an extension to the ISO C standard.
561		Where applicable, functions are marked with the TMO margin legend in the SYNOPSIS section.
562		Where additional semantics apply to a function, the material is identified by use of the TMO margin legend.
563		

564	TMR	Timers
565		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
567		Where applicable, functions are marked with the TMR margin legend in the SYNOPSIS section.
568		Where additional semantics apply to a function, the material is identified by use of the TMR margin legend.
570	TPI	Thread Priority Inheritance
571		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
573		Where applicable, functions are marked with the TPI margin legend in the SYNOPSIS section.
574		Where additional semantics apply to a function, the material is identified by use of the TPI margin legend.
576	TPP	Thread Priority Protection
577		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
579		Where applicable, functions are marked with the TPP margin legend in the SYNOPSIS section.
580		Where additional semantics apply to a function, the material is identified by use of the TPP margin legend.
582	TPS	Thread Execution Scheduling
583		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
585		Where applicable, functions are marked with the TPS margin legend for the SYNOPSIS section.
586		Where additional semantics apply to a function, the material is identified by use of the TPS margin legend.
588	TRC	Trace
589		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
591		Where applicable, functions are marked with the TRC margin legend in the SYNOPSIS section.
592		Where additional semantics apply to a function, the material is identified by use of the TRC margin legend.
594	TRI	Trace Inherit
595		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
597		Where applicable, functions are marked with the TRI margin legend in the SYNOPSIS section.
598		Where additional semantics apply to a function, the material is identified by use of the TRI margin legend.
600	TRL	Trace Log
601		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
603		Where applicable, functions are marked with the TRL margin legend in the SYNOPSIS section.
604		Where additional semantics apply to a function, the material is identified by use of the TRL margin legend.
606	TSA	Thread Stack Address Attribute
607		The functionality described is optional. The functionality described is also an extension to the ISO C standard.
608		

609	Where applicable, functions are marked with the TSA margin legend for the SYNOPSIS section.
610	Where additional semantics apply to a function, the material is identified by use of the TSA margin legend.
612	TSF Thread-Safe Functions
613	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
615	Where applicable, functions are marked with the TSF margin legend in the SYNOPSIS section.
616	Where additional semantics apply to a function, the material is identified by use of the TSF margin legend.
618	TSH Thread Process-Shared Synchronization
619	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
621	Where applicable, functions are marked with the TSH margin legend in the SYNOPSIS section.
622	Where additional semantics apply to a function, the material is identified by use of the TSH margin legend.
624	TSP Thread Sporadic Server
625	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
627	Where applicable, functions are marked with the TSP margin legend in the SYNOPSIS section.
628	Where additional semantics apply to a function, the material is identified by use of the TSP margin legend.
630	TSS Thread Stack Address Size
631	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
633	Where applicable, functions are marked with the TSS margin legend in the SYNOPSIS section.
634	Where additional semantics apply to a function, the material is identified by use of the TSS margin legend.
636	TYM Typed Memory Objects
637	The functionality described is optional. The functionality described is also an extension to the ISO C standard.
639	Where applicable, functions are marked with the TYM margin legend in the SYNOPSIS section.
640	Where additional semantics apply to a function, the material is identified by use of the TYM margin legend.
642	UP User Portability Utilities
643	The functionality described is optional.
644	Where applicable, utilities are marked with the UP margin legend in the SYNOPSIS section.
645	Where additional semantics apply to a utility, the material is identified by use of the UP margin legend.
647	XSI Extension
648	The functionality described is an XSI extension. Functionality marked XSI is also an extension to the ISO C standard. Application writers may confidently make use of an extension on all systems supporting the X/Open System Interfaces Extension.
651	If an entire SYNOPSIS section is shaded and marked XSI, all the functionality described in that reference page is an extension. See the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.439, XSI.
653	

654 XSR XSI STREAMS
 655 The functionality described is optional. The functionality described is also an extension to the
 656 ISO C standard.
 657 Where applicable, functions are marked with the XSR margin legend in the SYNOPSIS section.
 658 Where additional semantics apply to a function, the material is identified by use of the XSR
 659 margin legend.

660 1.9 Utility Limits

661 This section lists magnitude limitations imposed by a specific implementation. The braces
 662 notation, {LIMIT}, is used in this volume of IEEE Std 1003.1-2001 to indicate these values, but the
 663 braces are not part of the name.

664 **Table 1-3 Utility Limit Minimum Values**

Name	Description	Value
{POSIX2_BC_BASE_MAX}	The maximum <i>obase</i> value allowed by the <i>bc</i> utility.	99
{POSIX2_BC_DIM_MAX}	The maximum number of elements permitted in an array by the <i>bc</i> utility.	2 048
{POSIX2_BC_SCALE_MAX}	The maximum <i>scale</i> value allowed by the <i>bc</i> utility.	99
{POSIX2_BC_STRING_MAX}	The maximum length of a string constant accepted by the <i>bc</i> utility.	1 000
{POSIX2_COLL_WEIGHTS_MAX}	The maximum number of weights that can be assigned to an entry of the LC_COLLATE order keyword in the locale definition file; see the border_start keyword in the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3.2, LC_COLLATE.	2
{POSIX2_EXPR_NEST_MAX}	The maximum number of expressions that can be nested within parentheses by the <i>expr</i> utility.	32
{POSIX2_LINE_MAX}	Unless otherwise noted, the maximum length, in bytes, of the input line of a utility (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing <newline>.	2 048
{POSIX2_RE_DUP_MAX}	The maximum number of repeated occurrences of a BRE permitted when using the interval notation \{m,n\}; see the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3.6, BREs Matching Multiple Characters.	255
{POSIX2_VERSION}	This value indicates the version of the utilities in this volume of IEEE Std 1003.1-2001 that are provided by the implementation. It changes with each published version.	200112L

697 The values specified in Table 1-3 represent the lowest values conforming implementations shall
 698 provide and, consequently, the largest values on which an application can rely without further

enquiries, as described below. These values shall be accessible to applications via the *getconf* utility (see *getconf* (on page 481)) and through the *sysconf()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001. The literal names shown in Table 1-3 (on page 17) apply only to the *getconf* utility; the high-level language binding describes the exact form of each name to be used by the interfaces in that binding.

Implementations may provide more liberal, or less restrictive, values than shown in Table 1-3 (on page 17). These possibly more liberal values are accessible using the symbols in Table 1-4.

The *sysconf()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001 or the *getconf* utility return the value of each symbol on each specific implementation. The value so retrieved is the largest, or most liberal, value that is available throughout the session lifetime, as determined at session creation. The literal names shown in the table apply only to the *getconf* utility; the high-level language binding describes the exact form of each name to be used by the interfaces in that binding.

All numeric limits defined by the System Interfaces volume of IEEE Std 1003.1-2001, such as {PATH_MAX}, shall also apply to this volume of IEEE Std 1003.1-2001. All the utilities defined by this volume of IEEE Std 1003.1-2001 are implicitly limited by these values, unless otherwise noted in the utility descriptions.

It is not guaranteed that the application can actually reach the specified limit of an implementation in any given case, or at all, as a lack of virtual memory or other resources may prevent this. The limit value indicates only that the implementation does not specifically impose any arbitrary, more restrictive limit.

Table 1-4 Symbolic Utility Limits

Name	Description	Minimum Value
{BC_BASE_MAX}	The maximum <i>obase</i> value allowed by the <i>bc</i> utility.	{POSIX2_BC_BASE_MAX}
{BC_DIM_MAX}	The maximum number of elements permitted in an array by the <i>bc</i> utility.	{POSIX2_BC_DIM_MAX}
{BC_SCALE_MAX}	The maximum <i>scale</i> value allowed by the <i>bc</i> utility.	{POSIX2_BC_SCALE_MAX}
{BC_STRING_MAX}	The maximum length of a string constant accepted by the <i>bc</i> utility.	{POSIX2_BC_STRING_MAX}
{COLL_WEIGHTS_MAX}	The maximum number of weights that can be assigned to an entry of the <i>LC_COLLATE</i> order keyword in the locale definition file; see the order_start keyword in the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3.2, <i>LC_COLLATE</i> .	{POSIX2_COLL_WEIGHTS_MAX}

Name	Description	Minimum Value
{EXPR_NEST_MAX}	The maximum number of expressions that can be nested within parentheses by the <i>expr</i> utility.	{POSIX2_EXPR_NEST_MAX}
{LINE_MAX}	Unless otherwise noted, the maximum length, in bytes, of the input line of a utility (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing <newline>.	{POSIX2_LINE_MAX}
{RE_DUP_MAX}	The maximum number of repeated occurrences of a BRE permitted when using the interval notation $\{m,n\}$; see the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3.6, BREs Matching Multiple Characters.	{POSIX2_RE_DUP_MAX}

The following value may be a constant within an implementation or may vary from one pathname to another.

{POSIX2_SYMLINKS}

When referring to a directory, the system supports the creation of symbolic links within that directory; for non-directory files, the meaning of {POSIX2_SYMLINKS} is undefined.

1.10 Grammar Conventions

Portions of this volume of IEEE Std 1003.1-2001 are expressed in terms of a special grammar notation. It is used to portray the complex syntax of certain program input. The grammar is based on the syntax used by the *yacc* utility. However, it does not represent fully functional *yacc* input, suitable for program use; the lexical processing and all semantic requirements are described only in textual form. The grammar is not based on source used in any traditional implementation and has not been tested with the semantic code that would normally be required to accompany it. Furthermore, there is no implication that the partial *yacc* code presented represents the most efficient, or only, means of supporting the complex syntax within the utility. Implementations may use other programming languages or algorithms, as long as the syntax supported is the same as that represented by the grammar.

The following typographical conventions are used in the grammar; they have no significance except to aid in reading.

- The identifiers for the reserved words of the language are shown with a leading capital letter. (These are terminals in the grammar; for example, **While**, **Case**.)

- 788 • The identifiers for terminals in the grammar are all named with uppercase letters and
789 underscores; for example, NEWLINE, ASSIGN_OP, NAME.
790 • The identifiers for non-terminals are all lowercase.

791 **1.11 Utility Description Defaults**

792 This section describes all of the subsections used within the utility descriptions, including:

- 793 • Intended usage of the section
794 • Global defaults that affect all the standard utilities
795 • The meanings of notations used in this volume of IEEE Std 1003.1-2001 that are specific to
796 individual utility sections

797 **NAME**

798 This section gives the name or names of the utility and briefly states its purpose.

799 **SYNOPSIS**

800 The SYNOPSIS section summarizes the syntax of the calling sequence for the utility,
801 including options, option-arguments, and operands. Standards for utility naming are
802 described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility
803 Syntax Guidelines; for describing the utility's arguments in the Base Definitions volume
804 of IEEE Std 1003.1-2001, Section 12.1, Utility Argument Syntax.

805 **DESCRIPTION**

806 The DESCRIPTION section describes the actions of the utility. If the utility has a very
807 complex set of subcommands or its own procedural language, an EXTENDED
808 DESCRIPTION section is also provided. Most explanations of optional functionality are
809 omitted here, as they are usually explained in the OPTIONS section.

810 As stated in Section 1.7.1.11 (on page 7), some functions are described in terms of
811 equivalent functionality. When specific functions are cited, the implementation shall
812 provide equivalent functionality including side effects associated with successful
813 execution of the function. The treatment of errors and intermediate results from the
814 individual functions cited is generally not specified by this volume of
815 IEEE Std 1003.1-2001. See the utility's EXIT STATUS and CONSEQUENCES OF
816 ERRORS sections for all actions associated with errors encountered by the utility.

817 **OPTIONS**

818 The OPTIONS section describes the utility options and option-arguments, and how
819 they modify the actions of the utility. Standard utilities that have options either fully
820 comply with the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility
821 Syntax Guidelines or describe all deviations. Apparent disagreements between
822 functionality descriptions in the OPTIONS and DESCRIPTION (or EXTENDED
823 DESCRIPTION) sections are always resolved in favor of the OPTIONS section.

824 Each OPTIONS section that uses the phrase "The ... utility shall conform to the Utility
825 Syntax Guidelines ..." refers only to the use of the utility as specified by this volume of
826 IEEE Std 1003.1-2001; implementation extensions should also conform to the
827 guidelines, but may allow exceptions for historical practice.

828 Unless otherwise stated in the utility description, when given an option unrecognized
829 by the implementation, or when a required option-argument is not provided, standard
830 utilities shall issue a diagnostic message to standard error and exit with a non-zero exit
831 status.

832 All utilities in this volume of IEEE Std 1003.1-2001 shall be capable of processing
833 arguments using eight-bit transparency.

834 **Default Behavior:** When this section is listed as “None.”, it means that the
835 implementation need not support any options. Standard utilities that do not accept
836 options, but that do accept operands, shall recognize “--” as a first argument to be
837 discarded.

838 The requirement for recognizing “--” is because conforming applications need a way
839 to shield their operands from any arbitrary options that the implementation may
840 provide as an extension. For example, if the standard utility *foo* is listed as taking no
841 options, and the application needed to give it a pathname with a leading hyphen, it
842 could safely do it as:

843 *foo* -- *-myfile*

844 and avoid any problems with **-m** used as an extension.

845 OPERANDS

846 The OPERANDS section describes the utility operands, and how they affect the actions
847 of the utility. Apparent disagreements between functionality descriptions in the
848 OPERANDS and DESCRIPTION (or EXTENDED DESCRIPTION) sections shall be
849 resolved in favor of the OPERANDS section.

850 If an operand naming a file can be specified as ‘-’, which means to use the standard
851 input instead of a named file, this is explicitly stated in this section. Unless otherwise
852 stated, the use of multiple instances of ‘-’ to mean standard input in a single
853 command produces unspecified results.

854 Unless otherwise stated, the standard utilities that accept operands shall process those
855 operands in the order specified in the command line.

856 **Default Behavior:** When this section is listed as “None.”, it means that the
857 implementation need not support any operands.

858 STDIN

859 The STDIN section describes the standard input of the utility. This section is frequently
860 merely a reference to the following section, as many utilities treat standard input and
861 input files in the same manner. Unless otherwise stated, all restrictions described in the
862 INPUT FILES section shall apply to this section as well.

863 Use of a terminal for standard input can cause any of the standard utilities that read
864 standard input to stop when used in the background. For this reason, applications
865 should not use interactive features in scripts to be placed in the background.

866 The specified standard input format of the standard utilities shall not depend on the
867 existence or value of the environment variables defined in this volume of
868 IEEE Std 1003.1-2001, except as provided by this volume of IEEE Std 1003.1-2001.

869 **Default Behavior:** When this section is listed as “Not used.”, it means that the
870 standard input shall not be read when the utility is used as described by this volume of
871 IEEE Std 1003.1-2001.

872 INPUT FILES

873 The INPUT FILES section describes the files, other than the standard input, used as
874 input by the utility. It includes files named as operands and option-arguments as well
875 as other files that are referred to, such as start-up and initialization files, databases, and
876 so on. Commonly-used files are generally described in one place and cross-referenced
877 by other utilities.

878 All utilities in this volume of IEEE Std 1003.1-2001 shall be capable of processing input
 879 files using eight-bit transparency.

880 When a standard utility reads a seekable input file and terminates without an error
 881 before it reaches end-of-file, the utility shall ensure that the file offset in the open file
 882 description is properly positioned just past the last byte processed by the utility. For
 883 files that are not seekable, the state of the file offset in the open file description for that
 884 file is unspecified. A conforming application shall not assume that the following three
 885 commands are equivalent:

```
886     tail -n +2 file
  887     (sed -n 1q; cat) < file
  888     cat file | (sed -n 1q; cat)
```

889 The second command is equivalent to the first only when the file is seekable. The third
 890 command leaves the file offset in the open file description in an unspecified state. Other
 891 utilities, such as *head*, *read*, and *sh*, have similar properties.

892 Some of the standard utilities, such as filters, process input files a line or a block at a
 893 time and have no restrictions on the maximum input file size. Some utilities may have
 894 size limitations that are not as obvious as file space or memory limitations. Such
 895 limitations should reflect resource limitations of some sort, not arbitrary limits set by
 896 implementors. Implementations shall document those utilities that are limited by
 897 constraints other than file system space, available memory, and other limits specifically
 898 cited by this volume of IEEE Std 1003.1-2001, and identify what the constraint is and
 899 indicate a way of estimating when the constraint would be reached. Similarly, some
 900 utilities descend the directory tree (recursively). Implementations shall also document
 901 any limits that they may have in descending the directory tree that are beyond limits
 902 cited by this volume of IEEE Std 1003.1-2001.

903 When an input file is described as a “text file”, the utility produces undefined results if
 904 given input that is not from a text file, unless otherwise stated. Some utilities (for
 905 example, *make*, *read*, *sh*) allow for continued input lines using an escaped <newline>
 906 convention; unless otherwise stated, the utility need not be able to accumulate more
 907 than {LINE_MAX} bytes from a set of multiple, continued input lines. Thus, for a
 908 conforming application the total of all the continued lines in a set cannot exceed
 909 {LINE_MAX}. If a utility using the escaped <newline> convention detects an end-of-
 910 file condition immediately after an escaped <newline>, the results are unspecified.

911 Record formats are described in a notation similar to that used by the C-language
 912 function, *printf()*. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5,
 913 File Format Notation for a description of this notation. The format description is
 914 intended to be sufficiently rigorous to allow other applications to generate these input
 915 files. However, since <blank>s can legitimately be included in some of the fields
 916 described by the standard utilities, particularly in locales other than the POSIX locale,
 917 this intent is not always realized.

918 **Default Behavior:** When this section is listed as “None.”, it means that no input files
 919 are required to be supplied when the utility is used as described by this volume of
 920 IEEE Std 1003.1-2001.

921 ENVIRONMENT VARIABLES

922 The ENVIRONMENT VARIABLES section lists what variables affect the utility’s
 923 execution.

924 The entire manner in which environment variables described in this volume of
 925 IEEE Std 1003.1-2001 affect the behavior of each utility is described in the

926 ENVIRONMENT VARIABLES section for that utility, in conjunction with the global
927 XSI effects of the *LANG*, *LC_ALL*, and *NLSPATH* environment variables described in the
928 Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
929 The existence or value of environment variables described in this volume of
930 IEEE Std 1003.1-2001 shall not otherwise affect the specified behavior of the standard
931 utilities. Any effects of the existence or value of environment variables not described by
932 this volume of IEEE Std 1003.1-2001 upon the standard utilities are unspecified.

933 For those standard utilities that use environment variables as a means for selecting a
934 utility to execute (such as *CC* in *make*), the string provided to the utility is subjected to
935 the path search described for *PATH* in the Base Definitions volume of
936 IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

937 All utilities in this volume of IEEE Std 1003.1-2001 shall be capable of processing
938 environment variable names and values using eight-bit transparency.

939 **Default Behavior:** When this section is listed as “None.”, it means that the behavior of
940 the utility is not directly affected by environment variables described by this volume of
941 IEEE Std 1003.1-2001 when the utility is used as described by this volume of
942 IEEE Std 1003.1-2001.

943 ASYNCHRONOUS EVENTS

944 The ASYNCHRONOUS EVENTS section lists how the utility reacts to such events as
945 signals and what signals are caught.

946 **Default Behavior:** When this section is listed as “Default.”, or it refers to “the standard
947 action for all other signals; see Section 1.11 (on page 20)” it means that the action taken
948 as a result of the signal shall be one of the following:

- 949 1. The action shall be that inherited from the parent according to the rules of
950 inheritance of signal actions defined in the System Interfaces volume of
951 IEEE Std 1003.1-2001.
- 952 2. When no action has been taken to change the default, the default action shall be
953 that specified by the System Interfaces volume of IEEE Std 1003.1-2001.
- 954 3. The result of the utility’s execution is as if default actions had been taken.

955 A utility is permitted to catch a signal, perform some additional processing (such as
956 deleting temporary files), restore the default signal action (or action inherited from the
957 parent process), and resignal itself.

958 STDOUT

959 The STDOUT section completely describes the standard output of the utility. This
960 section is frequently merely a reference to the following section, OUTPUT FILES,
961 because many utilities treat standard output and output files in the same manner.

962 Use of a terminal for standard output may cause any of the standard utilities that write
963 standard output to stop when used in the background. For this reason, applications
964 should not use interactive features in scripts to be placed in the background.

965 Record formats are described in a notation similar to that used by the C-language
966 function, *printf()*. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5,
967 File Format Notation for a description of this notation.

968 The specified standard output of the standard utilities shall not depend on the
969 existence or value of the environment variables defined in this volume of
970 IEEE Std 1003.1-2001, except as provided by this volume of IEEE Std 1003.1-2001.

971 Some of the standard utilities describe their output using the verb *display*, defined in
 972 the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.132, Display. Output
 973 described in the STDOUT sections of such utilities may be produced using means other
 974 than standard output. When standard output is directed to a terminal, the output
 975 described shall be written directly to the terminal. Otherwise, the results are undefined.

976 **Default Behavior:** When this section is listed as “Not used.”, it means that the
 977 standard output shall not be written when the utility is used as described by this
 978 volume of IEEE Std 1003.1-2001.

979 STDERR

980 The STDERR section describes the standard error output of the utility. Only those
 981 messages that are purposely sent by the utility are described.

982 Use of a terminal for standard error may cause any of the standard utilities that write
 983 standard error output to stop when used in the background. For this reason,
 984 applications should not use interactive features in scripts to be placed in the
 985 background.

986 The format of diagnostic messages for most utilities is unspecified, but the language
 987 and cultural conventions of diagnostic and informative messages whose format is
 988 unspecified by this volume of IEEE Std 1003.1-2001 should be affected by the setting of
 989 XSI *LC_MESSAGES* and *NLSPATH*.

990 The specified standard error output of standard utilities shall not depend on the
 991 existence or value of the environment variables defined in this volume of
 992 IEEE Std 1003.1-2001, except as provided by this volume of IEEE Std 1003.1-2001.

993 **Default Behavior:** When this section is listed as “The standard error shall be used only
 994 for diagnostic messages.”, it means that, unless otherwise stated, the diagnostic
 995 messages shall be sent to the standard error only when the exit status is non-zero and
 996 the utility is used as described by this volume of IEEE Std 1003.1-2001.

997 When this section is listed as “Not used.”, it means that the standard error shall not be
 998 used when the utility is used as described in this volume of IEEE Std 1003.1-2001.

999 OUTPUT FILES

1000 The OUTPUT FILES section completely describes the files created or modified by the
 1001 utility. Temporary or system files that are created for internal usage by this utility or
 1002 other parts of the implementation (for example, spool, log, and audit files) are not
 1003 described in this, or any, section. The utilities creating such files and the names of such
 1004 files are unspecified. If applications are written to use temporary or intermediate files,
 1005 they should use the *TMPDIR* environment variable, if it is set and represents an
 1006 accessible directory, to select the location of temporary files.

1007 Implementations shall ensure that temporary files, when used by the standard utilities,
 1008 are named so that different utilities or multiple instances of the same utility can operate
 1009 simultaneously without regard to their working directories, or any other process
 1010 characteristic other than process ID. There are two exceptions to this rule:

- 1011 1. Resources for temporary files other than the name space (for example, disk space,
 1012 available directory entries, or number of processes allowed) are not guaranteed.
- 1013 2. Certain standard utilities generate output files that are intended as input for other
 1014 utilities (for example, *lex* generates *lex.yy.c*), and these cannot have unique
 1015 names. These cases are explicitly identified in the descriptions of the respective
 1016 utilities.

1017 Any temporary file created by the implementation shall be removed by the
1018 implementation upon a utility's successful exit, exit because of errors, or before
1019 termination by any of the SIGHUP, SIGINT, or SIGTERM signals, unless specified
1020 otherwise by the utility description.

1021 Receipt of the SIGQUIT signal should generally cause termination (unless in some
1022 debugging mode) that would bypass any attempted recovery actions.

1023 Record formats are described in a notation similar to that used by the C-language
1024 function, *printf()*; see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5,
1025 File Format Notation for a description of this notation.

1026 **Default Behavior:** When this section is listed as "None.", it means that no files are
1027 created or modified as a consequence of direct action on the part of the utility when the
1028 utility is used as described by this volume of IEEE Std 1003.1-2001. However, the
1029 utility may create or modify system files, such as log files, that are outside the utility's
1030 normal execution environment.

1031 EXTENDED DESCRIPTION

1032 The EXTENDED DESCRIPTION section provides a place for describing the actions of
1033 very complicated utilities, such as text editors or language processors, which typically
1034 have elaborate command languages.

1035 **Default Behavior:** When this section is listed as "None.", no further description is
1036 necessary.

1037 EXIT STATUS

1038 The EXIT STATUS section describes the values the utility shall return to the calling
1039 program, or shell, and the conditions that cause these values to be returned. Usually,
1040 utilities return zero for successful completion and values greater than zero for various
1041 error conditions. If specific numeric values are listed in this section, the system shall
1042 use those values for the errors described. In some cases, status values are listed more
1043 loosely, such as >0. A strictly conforming application shall not rely on any specific
1044 value in the range shown and shall be prepared to receive any value in the range.

1045 For example, a utility may list zero as a successful return, 1 as a failure for a specific
1046 reason, and >1 as "an error occurred". In this case, unspecified conditions may cause a
1047 2 or 3, or other value, to be returned. A conforming application should be written so
1048 that it tests for successful exit status values (zero in this case), rather than relying upon
1049 the single specific error value listed in this volume of IEEE Std 1003.1-2001. In that
1050 way, it has maximum portability, even on implementations with extensions.

1051 Unspecified error conditions may be represented by specific values not listed in this
1052 volume of IEEE Std 1003.1-2001.

1053 CONSEQUENCES OF ERRORS

1054 The CONSEQUENCES OF ERRORS section describes the effects on the environment,
1055 file systems, process state, and so on, when error conditions occur. It does not describe
1056 error messages produced or exit status values used.

1057 The many reasons for failure of a utility are generally not specified by the utility
1058 descriptions. Utilities may terminate prematurely if they encounter: invalid usage of
1059 options, arguments, or environment variables; invalid usage of the complex syntaxes
1060 expressed in EXTENDED DESCRIPTION sections; difficulties accessing, creating,
1061 reading, or writing files; or difficulties associated with the privileges of the process.

1062 The following shall apply to each utility, unless otherwise stated:

- 1063 • If the requested action cannot be performed on an operand representing a file,
1064 directory, user, process, and so on, the utility shall issue a diagnostic message to
1065 standard error and continue processing the next operand in sequence, but the final
1066 exit status shall be returned as non-zero.

1067 For a utility that recursively traverses a file hierarchy (such as *find* or *chown -R*), if
1068 the requested action cannot be performed on a file or directory encountered in the
1069 hierarchy, the utility shall issue a diagnostic message to standard error and continue
1070 processing the remaining files in the hierarchy, but the final exit status shall be
1071 returned as non-zero.

- 1072 • If the requested action characterized by an option or option-argument cannot be
1073 performed, the utility shall issue a diagnostic message to standard error and the exit
1074 status returned shall be non-zero.
- 1075 • When an unrecoverable error condition is encountered, the utility shall exit with a
1076 non-zero exit status.
- 1077 • A diagnostic message shall be written to standard error whenever an error
1078 condition occurs.

1079 When a utility encounters an error condition several actions are possible, depending on
1080 the severity of the error and the state of the utility. Included in the possible actions of
1081 various utilities are: deletion of temporary or intermediate work files; deletion of
1082 incomplete files; validity checking of the file system or directory.

1083 **Default Behavior:** When this section is listed as “Default.”, it means that any changes
1084 to the environment are unspecified.

1085 APPLICATION USAGE

1086 This section is informative.

1087 The APPLICATION USAGE section gives advice to the application programmer or user
1088 about the way the utility should be used.

1089 EXAMPLES

1090 This section is informative.

1091 The EXAMPLES section gives one or more examples of usage, where appropriate. In
1092 the event of conflict between an example and a normative part of the specification, the
1093 normative material is to be taken as correct.

1094 In all examples, quoting has been used, showing how sample commands (utility names
1095 combined with arguments) could be passed correctly to a shell (see *sh*) or as a string to
1096 the *system()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001.
1097 Such quoting would not be used if the utility is invoked using one of the *exec* functions
1098 defined in the System Interfaces volume of IEEE Std 1003.1-2001.

1099 RATIONALE

1100 This section is informative.

1101 This section contains historical information concerning the contents of this volume of
1102 IEEE Std 1003.1-2001 and why features were included or discarded by the standard
1103 developers.

1104 FUTURE DIRECTIONS

1105 This section is informative.

1106 The FUTURE DIRECTIONS section should be used as a guide to current thinking; there
1107 is not necessarily a commitment to implement all of these future directions in their

1108 entirety.

1109 **SEE ALSO**

1110 This section is informative.

1111 The SEE ALSO section lists related entries.

1112 **CHANGE HISTORY**

1113 This section is informative.

1114 This section shows the derivation of the entry and any significant changes that have
1115 been made to it.

1116 Certain of the standard utilities describe how they can invoke other utilities or applications, such
1117 as by passing a command string to the command interpreter. The external influences (STDIN,
1118 ENVIRONMENT VARIABLES, and so on) and external effects (STDOUT, CONSEQUENCES OF
1119 ERRORS, and so on) of such invoked utilities are not described in the section concerning the
1120 standard utility that invokes them.

1121 **1.12 Considerations for Utilities in Support of Files of Arbitrary Size**

1122 The following utilities support files of any size up to the maximum that can be created by the
1123 implementation. This support includes correct writing of file size-related values (such as file
1124 sizes and offsets, line numbers, and block counts) and correct interpretation of command line
1125 arguments that contain such values.

1126	<i>basename</i>	Return non-directory portion of pathname.
1127	<i>cat</i>	Concatenate and print files.
1128	<i>cd</i>	Change working directory.
1129	<i>chgrp</i>	Change file group ownership.
1130	<i>chmod</i>	Change file modes.
1131	<i>chown</i>	Change file ownership.
1132	<i>cksum</i>	Write file checksums and sizes.
1133	<i>cmp</i>	Compare two files.
1134	<i>cp</i>	Copy files.
1135	<i>dd</i>	Convert and copy a file.
1136	<i>df</i>	Report free disk space.
1137	<i>dirname</i>	Return directory portion of pathname.
1138	<i>du</i>	Estimate file space usage.
1139	<i>find</i>	Find files.
1140	<i>ln</i>	Link files.
1141	<i>ls</i>	List directory contents.
1142	<i>mkdir</i>	Make directories.
1143	<i>mv</i>	Move files.

1144	<i>pathchk</i>	Check pathnames.
1145	<i>pwd</i>	Return working directory name.
1146	<i>rm</i>	Remove directory entries.
1147	<i>rmdir</i>	Remove directories.
1148	<i>sh</i>	Shell, the standard command language interpreter.
1149	<i>sum</i>	Print checksum and block or byte count of a file.
1150	<i>test</i>	Evaluate expression.
1151	<i>touch</i>	Change file access and modification times.
1152	<i>ulimit</i>	Set or report file size limit.
1153	Exceptions to the requirement that utilities support files of any size up to the maximum are as follows:	
1154		
1155	1.	Uses of files as command scripts, or for configuration or control, are exempt. For example, it is not required that <i>sh</i> be able to read an arbitrarily large .profile .
1156	2.	Shell input and output redirection are exempt. For example, it is not required that the redirections <i>sum < file</i> or <i>echo foo > file</i> succeed for an arbitrarily large existing file.
1157		
1158		

1159 1.13 Built-In Utilities

1160 Any of the standard utilities may be implemented as regular built-in utilities within the
 1161 command language interpreter. This is usually done to increase the performance of frequently
 1162 used utilities or to achieve functionality that would be more difficult in a separate environment.
 1163 The utilities named in Table 1-5 are frequently provided in built-in form. All of the utilities
 1164 named in the table have special properties in terms of command search order within the shell, as
 1165 described in Section 2.9.1.1 (on page 48).

1166 **Table 1-5** Regular Built-In Utilities

1167	<i>alias</i>	<i>false</i>	<i>jobs</i>	<i>read</i>	<i>wait</i>
1168	<i>bg</i>	<i>fc</i>	<i>kill</i>	<i>true</i>	
1169	<i>cd</i>	<i>fg</i>	<i>newgrp</i>	<i>umask</i>	
1170	<i>command</i>	<i>getopts</i>	<i>pwd</i>	<i>unalias</i>	

1171 However, all of the standard utilities, including the regular built-ins in the table, but not the
 1172 special built-ins described in Section 2.14 (on page 64), shall be implemented in a manner so that
 1173 they can be accessed via the *exec* family of functions as defined in the System Interfaces volume
 1174 of IEEE Std 1003.1-2001 and can be invoked directly by those standard utilities that require it
 1175 (*env*, *find*, *nice*, *nohup*, *time*, *xargs*).

Shell Command Language

1177

This chapter contains the definition of the Shell Command Language.

1178

2.1 Shell Introduction

1179 The shell is a command language interpreter. This chapter describes the syntax of that command
 1180 language as it is used by the *sh* utility and the *system()* and *popen()* functions defined in the
 1181 System Interfaces volume of IEEE Std 1003.1-2001.

1182 The shell operates according to the following general overview of operations. The specific
 1183 details are included in the cited sections of this chapter.

- 1184 1. The shell reads its input from a file (see *sh*), from the *-c* option or from the *system()* and
 1185 *popen()* functions defined in the System Interfaces volume of IEEE Std 1003.1-2001. If the
 1186 first line of a file of shell commands starts with the characters "#!", the results are
 1187 unspecified.
- 1188 2. The shell breaks the input into tokens: words and operators; see Section 2.3 (on page 31).
- 1189 3. The shell parses the input into simple commands (see Section 2.9.1 (on page 47)) and
 1190 compound commands (see Section 2.9.4 (on page 52)).
- 1191 4. The shell performs various expansions (separately) on different parts of each command,
 1192 resulting in a list of pathnames and fields to be treated as a command and arguments; see
 1193 Section 2.6 (on page 36).
- 1194 5. The shell performs redirection (see Section 2.7 (on page 43)) and removes redirection
 1195 operators and their operands from the parameter list.
- 1196 6. The shell executes a function (see Section 2.9.5 (on page 54)), built-in (see Section 2.14 (on
 1197 page 64)), executable file, or script, giving the names of the arguments as positional
 1198 parameters numbered 1 to *n*, and the name of the command (or in the case of a function
 1199 within a script, the name of the script) as the positional parameter numbered 0 (see Section
 1200 2.9.1.1 (on page 48)).
- 1201 7. The shell optionally waits for the command to complete and collects the exit status (see
 1202 Section 2.8.2 (on page 46)).

1203 **2.2 Quoting**

1204 Quoting is used to remove the special meaning of certain characters or words to the shell.
 1205 Quoting can be used to preserve the literal meaning of the special characters in the next
 1206 paragraph, prevent reserved words from being recognized as such, and prevent parameter
 1207 expansion and command substitution within here-document processing (see Section 2.7.4 (on
 1208 page 44)).

1209 The application shall quote the following characters if they are to represent themselves:

1210 | & ; < > () \$ ` \ " ' <space> <tab> <newline>

1211 and the following may need to be quoted under certain circumstances. That is, these characters
 1212 may be special depending on conditions described elsewhere in this volume of
 1213 IEEE Std 1003.1-2001:

1214 * ? [# ~ = %

1215 The various quoting mechanisms are the escape character, single-quotes, and double-quotes.
 1216 The here-document represents another form of quoting; see Section 2.7.4 (on page 44).

1217 **2.2.1 Escape Character (Backslash)**

1218 A backslash that is not quoted shall preserve the literal value of the following character, with the
 1219 exception of a <newline>. If a <newline> follows the backslash, the shell shall interpret this as
 1220 line continuation. The backslash and <newline>s shall be removed before splitting the input into
 1221 tokens. Since the escaped <newline> is removed entirely from the input and is not replaced by
 1222 any white space, it cannot serve as a token separator.

1223 **2.2.2 Single-Quotes**

1224 Enclosing characters in single-quotes (‘ ’) shall preserve the literal value of each character
 1225 within the single-quotes. A single-quote cannot occur within single-quotes.

1226 **2.2.3 Double-Quotes**

1227 Enclosing characters in double-quotes (" ") shall preserve the literal value of all characters
 1228 within the double-quotes, with the exception of the characters dollar sign, backquote, and
 1229 backslash, as follows:

1230 \$ The dollar sign shall retain its special meaning introducing parameter expansion (see
 1231 Section 2.6.2 (on page 37)), a form of command substitution (see Section 2.6.3 (on page 40)),
 1232 and arithmetic expansion (see Section 2.6.4 (on page 41)).

1233 The input characters within the quoted string that are also enclosed between "\$(" and the
 1234 matching ")" shall not be affected by the double-quotes, but rather shall define that
 1235 command whose output replaces the "\$(...)" when the word is expanded. The
 1236 tokenizing rules in Section 2.3 (on page 31), not including the alias substitutions in Section
 1237 2.3.1 (on page 32), shall be applied recursively to find the matching ') '.

1238 Within the string of characters from an enclosed "\${" to the matching ' } ', an even number
 1239 of unescaped double-quotes or single-quotes, if any, shall occur. A preceding backslash
 1240 character shall be used to escape a literal '{' or '} '. The rule in Section 2.6.2 (on page 37)
 1241 shall be used to determine the matching ' } '.

1242 ' The backquote shall retain its special meaning introducing the other form of command
 1243 substitution (see Section 2.6.3 (on page 40)). The portion of the quoted string from the initial
 1244 backquote and the characters up to the next backquote that is not preceded by a backslash,

1245 having escape characters removed, defines that command whose output replaces ``...''
 1246 when the word is expanded. Either of the following cases produces undefined results:

- 1247 • A single-quoted or double-quoted string that begins, but does not end, within the
 1248 ``...'' sequence
 - 1249 • A ``...'' sequence that begins, but does not end, within the same double-quoted
 1250 string
- 1251 \ The backslash shall retain its special meaning as an escape character (see Section 2.2.1 (on
 1252 page 30)) only when followed by one of the following characters when considered special:

1253 \$ ` " \ <newline>

1254 The application shall ensure that a double-quote is preceded by a backslash to be included
 1255 within double-quotes. The parameter '@' has special meaning inside double-quotes and is
 1256 described in Section 2.5.2 (on page 34).

1257 2.3 Token Recognition

1258 The shell shall read its input in terms of lines from a file, from a terminal in the case of an
 1259 interactive shell, or from a string in the case of *sh -c* or *system()*. The input lines can be of
 1260 unlimited length. These lines shall be parsed using two major modes: ordinary token recognition
 1261 and processing of here-documents.

1262 When an **io_here** token has been recognized by the grammar (see Section 2.10 (on page 55)), one
 1263 or more of the subsequent lines immediately following the next **NEWLINE** token form the body
 1264 of one or more here-documents and shall be parsed according to the rules of Section 2.7.4 (on
 1265 page 44).

1266 When it is not processing an **io_here**, the shell shall break its input into tokens by applying the
 1267 first applicable rule below to the next character in its input. The token shall be from the current
 1268 position in the input until a token is delimited according to one of the rules below; the characters
 1269 forming the token are exactly those in the input, including any quoting characters. If it is
 1270 indicated that a token is delimited, and no characters have been included in a token, processing
 1271 shall continue until an actual token is delimited.

- 1272 1. If the end of input is recognized, the current token shall be delimited. If there is no current
 1273 token, the end-of-input indicator shall be returned as the token.
- 1274 2. If the previous character was used as part of an operator and the current character is not
 1275 quoted and can be used with the current characters to form an operator, it shall be used as
 1276 part of that (operator) token.
- 1277 3. If the previous character was used as part of an operator and the current character cannot
 1278 be used with the current characters to form an operator, the operator containing the
 1279 previous character shall be delimited.
- 1280 4. If the current character is backslash, single-quote, or double-quote ('\'', ''', or '') and
 1281 it is not quoted, it shall affect quoting for subsequent characters up to the end of the quoted
 1282 text. The rules for quoting are as described in Section 2.2 (on page 30). During token
 1283 recognition no substitutions shall be actually performed, and the result token shall contain
 1284 exactly the characters that appear in the input (except for <newline> joining), unmodified,
 1285 including any embedded or enclosing quotes or substitution operators, between the quote
 1286 mark and the end of the quoted text. The token shall not be delimited by the end of the
 1287 quoted field.

- 1288 5. If the current character is an unquoted '\$' or ' ', the shell shall identify the start of any
1289 candidates for parameter expansion (Section 2.6.2 (on page 37)), command substitution
1290 (Section 2.6.3 (on page 40)), or arithmetic expansion (Section 2.6.4 (on page 41)) from their
1291 introductory unquoted character sequences: '\$' or "\${", "\$(" or ' ', and "\$((", respectively. The shell shall read sufficient input to determine the end of the unit to be
1292 expanded (as explained in the cited sections). While processing the characters, if instances
1293 of expansions or quoting are found nested within the substitution, the shell shall
1294 recursively process them in the manner specified for the construct that is found. The
1295 characters found from the beginning of the substitution to its end, allowing for any
1296 recursion necessary to recognize embedded constructs, shall be included unmodified in the
1297 result token, including any embedded or enclosing substitution operators or quotes. The
1298 token shall not be delimited by the end of the substitution.
1299
- 1300 6. If the current character is not quoted and can be used as the first character of a new
1301 operator, the current token (if any) shall be delimited. The current character shall be used
1302 as the beginning of the next (operator) token.
1303 7. If the current character is an unquoted <newline>, the current token shall be delimited.
1304 8. If the current character is an unquoted <blank>, any token containing the previous
1305 character is delimited and the current character shall be discarded.
1306 9. If the previous character was part of a word, the current character shall be appended to
1307 that word.
1308 10. If the current character is a '#', it and all subsequent characters up to, but excluding, the
1309 next <newline> shall be discarded as a comment. The <newline> that ends the line is not
1310 considered part of the comment.
1311 11. The current character is used as the start of a new word.

1312 Once a token is delimited, it is categorized as required by the grammar in Section 2.10 (on page
1313 55).

2.3.1 Alias Substitution

1315 UP XSI The processing of aliases shall be supported on all XSI-conformant systems or if the system
1316 supports the User Portability Utilities option (and the rest of this section is not further shaded for
1317 these options).

1318 After a token has been delimited, but before applying the grammatical rules in Section 2.10 (on
1319 page 55), a resulting word that is identified to be the command name word of a simple
1320 command shall be examined to determine whether it is an unquoted, valid alias name. However,
1321 reserved words in correct grammatical context shall not be candidates for alias substitution. A
1322 valid alias name (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.10, Alias
1323 Name) shall be one that has been defined by the *alias* utility and not subsequently undefined
1324 using *unalias*. Implementations also may provide predefined valid aliases that are in effect when
1325 the shell is invoked. To prevent infinite loops in recursive aliasing, if the shell is not currently
1326 processing an alias of the same name, the word shall be replaced by the value of the alias;
1327 otherwise, it shall not be replaced.

1328 If the value of the alias replacing the word ends in a <blank>, the shell shall check the next
1329 command word for alias substitution; this process shall continue until a word is found that is not
1330 a valid alias or an alias value does not end in a <blank>.

1331 When used as specified by this volume of IEEE Std 1003.1-2001, alias definitions shall not be
1332 inherited by separate invocations of the shell or by the utility execution environments invoked
1333 by the shell; see Section 2.12 (on page 61).

1334 2.4 Reserved Words

1335 Reserved words are words that have special meaning to the shell; see Section 2.9 (on page 47).
 1336 The following words shall be recognized as reserved words:

1337 !	1338 do	1339 esac	1340 in
{	done	fi	then
}	elif	for	until
case	else	if	while

1341 This recognition shall only occur when none of the characters is quoted and when the word is
 1342 used as:

- 1343 • The first word of a command
- 1344 • The first word following one of the reserved words other than **case**, **for**, or **in**
- 1345 • The third word in a **case** command (only **in** is valid in this case)
- 1346 • The third word in a **for** command (only **in** and **do** are valid in this case)

1347 See the grammar in Section 2.10 (on page 55).

1348 The following words may be recognized as reserved words on some implementations (when
 1349 none of the characters are quoted), causing unspecified results:

1350 [1351]	1352 function	1353 select
---------------	---------------	----------------------	--------------------

1351 Words that are the concatenation of a name and a colon (' : ') are reserved; their use produces
 1352 unspecified results.

1353 2.5 Parameters and Variables

1354 A parameter can be denoted by a name, a number, or one of the special characters listed in
 1355 Section 2.5.2 (on page 34). A variable is a parameter denoted by a name.

1356 A parameter is set if it has an assigned value (null is a valid value). Once a variable is set, it can
 1357 only be unset by using the *unset* special built-in command.

1358 2.5.1 Positional Parameters

1359 A positional parameter is a parameter denoted by the decimal value represented by one or more
 1360 digits, other than the single digit 0. The digits denoting the positional parameters shall always be
 1361 interpreted as a decimal value, even if there is a leading zero. When a positional parameter with
 1362 more than one digit is specified, the application shall enclose the digits in braces (see Section
 1363 2.6.2 (on page 37)). Positional parameters are initially assigned when the shell is invoked (see
 1364 *sh*), temporarily replaced when a shell function is invoked (see Section 2.9.5 (on page 54)), and
 1365 can be reassigned with the *set* special built-in command.

1366 **2.5.2 Special Parameters**

1367 Listed below are the special parameters and the values to which they shall expand. Only the
 1368 values of the special parameters are listed; see Section 2.6 (on page 36) for a detailed summary of
 1369 all the stages involved in expanding words.

- 1370 @ Expands to the positional parameters, starting from one. When the expansion occurs within
 1371 double-quotes, and where field splitting (see Section 2.6.5 (on page 42)) is performed, each
 1372 positional parameter shall expand as a separate field, with the provision that the expansion
 1373 of the first parameter shall still be joined with the beginning part of the original word
 1374 (assuming that the expanded parameter was embedded within a word), and the expansion
 1375 of the last parameter shall still be joined with the last part of the original word. If there are
 1376 no positional parameters, the expansion of '@' shall generate zero fields, even when '@' is
 1377 double-quoted.
- 1378 * Expands to the positional parameters, starting from one. When the expansion occurs within
 1379 a double-quoted string (see Section 2.2.3 (on page 30)), it shall expand to a single field with
 1380 the value of each parameter separated by the first character of the *IFS* variable, or by a
 1381 <space> if *IFS* is unset. If *IFS* is set to a null string, this is not equivalent to unsetting it; its
 1382 first character does not exist, so the parameter values are concatenated.
- 1383 # Expands to the decimal number of positional parameters. The command name (parameter
 1384 0) shall not be counted in the number given by '#' because it is a special parameter, not a
 1385 positional parameter.
- 1386 ? Expands to the decimal exit status of the most recent pipeline (see Section 2.9.2 (on page
 1387 49)).
- 1388 - (Hyphen.) Expands to the current option flags (the single-letter option names concatenated
 1389 into a string) as specified on invocation, by the *set* special built-in command, or implicitly by
 1390 the shell.
- 1391 \$ Expands to the decimal process ID of the invoked shell. In a subshell (see Section 2.12 (on
 1392 page 61)), '\$' shall expand to the same value as that of the current shell.
- 1393 ! Expands to the decimal process ID of the most recent background command (see Section
 1394 2.9.3 (on page 50)) executed from the current shell. (For example, background commands
 1395 executed from subshells do not affect the value of "\$!" in the current shell environment.)
 1396 For a pipeline, the process ID is that of the last command in the pipeline.
- 1397 0 (Zero.) Expands to the name of the shell or shell script. See *sh* (on page 847) for a detailed
 1398 description of how this name is derived.

1399 See the description of the *IFS* variable in Section 2.5.3.

1400 **2.5.3 Shell Variables**

1401 Variables shall be initialized from the environment (as defined by the Base Definitions volume of
 1402 IEEE Std 1003.1-2001, Chapter 8, Environment Variables and the *exec* function in the System
 1403 Interfaces volume of IEEE Std 1003.1-2001) and can be given new values with variable
 1404 assignment commands. If a variable is initialized from the environment, it shall be marked for
 1405 export immediately; see the *export* special built-in. New variables can be defined and initialized
 1406 with variable assignments, with the *read* or *getopts* utilities, with the *name* parameter in a *for*
 1407 loop, with the \${name=word} expansion, or with other mechanisms provided as implementation
 1408 extensions.

1409 The following variables shall affect the execution of the shell:

1410	UP XSI	<i>ENV</i>	The processing of the <i>ENV</i> shell variable shall be supported on all XSI-conformant systems or if the system supports the User Portability Utilities option.
1413			This variable, when and only when an interactive shell is invoked, shall be subjected to parameter expansion (see Section 2.6.2 (on page 37)) by the shell and the resulting value shall be used as a pathname of a file containing shell commands to execute in the current environment. The file need not be executable. If the expanded value of <i>ENV</i> is not an absolute pathname, the results are unspecified. <i>ENV</i> shall be ignored if the user's real and effective user IDs or real and effective group IDs are different.
1420		<i>HOME</i>	The pathname of the user's home directory. The contents of <i>HOME</i> are used in tilde expansion (see Section 2.6.1 (on page 37)).
1422		<i>IFS</i>	(Input Field Separators.) A string treated as a list of characters that is used for field splitting and to split lines into fields with the <i>read</i> command. If <i>IFS</i> is not set, the shell shall behave as if the value of <i>IFS</i> is <space>, <tab>, and <newline>; see Section 2.6.5 (on page 42). Implementations may ignore the value of <i>IFS</i> in the environment at the time the shell is invoked, treating <i>IFS</i> as if it were not set.
1428		<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
1432		<i>LC_ALL</i>	The value of this variable overrides the <i>LC_*</i> variables and <i>LANG</i> , as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
1435		<i>LC_COLLATE</i>	Determine the behavior of range expressions, equivalence classes, and multi-character collating elements within pattern matching.
1437		<i>LC_CTYPE</i>	Determine the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters), which characters are defined as letters (character class alpha) and <blank>s (character class blank), and the behavior of character classes within pattern matching. Changing the value of <i>LC_CTYPE</i> after the shell has started shall not affect the lexical processing of shell commands in the current shell execution environment or its subshells. Invoking a shell script or performing <i>exec sh</i> subjects the new shell to the changes in <i>LC_CTYPE</i> .
1445		<i>LC_MESSAGES</i>	Determine the language in which messages should be written.
1446		<i>LINENO</i>	Set by the shell to a decimal number representing the current sequential line number (numbered starting with 1) within a script or function before it executes each command. If the user unsets or resets <i>LINENO</i> , the variable may lose its special meaning for the life of the shell. If the shell is not currently executing a script or function, the value of <i>LINENO</i> is unspecified. This volume of IEEE Std 1003.1-2001 specifies the effects of the variable only for systems supporting the User Portability Utilities option.
1453	XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
1455		<i>PATH</i>	A string formatted as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables, used to effect

1457		command interpretation; see Section 2.9.1.1 (on page 48).
1458	<i>PPID</i>	Set by the shell to the decimal process ID of the process that invoked this shell.
1459		In a subshell (see Section 2.12 (on page 61)), <i>PPID</i> shall be set to the same value as that of the parent of the current shell. For example, <i>echo \$PPID</i> and (<i>echo \$PPID</i>) would produce the same value. This volume of IEEE Std 1003.1-2001 specifies the effects of the variable only for systems supporting the User Portability Utilities option.
1460		
1461		
1462		
1463		
1464	<i>PS1</i>	Each time an interactive shell is ready to read a command, the value of this variable shall be subjected to parameter expansion and written to standard error. The default value shall be " \$ ". For users who have specific additional implementation-defined privileges, the default may be another, implementation-defined value. The shell shall replace each instance of the character ' ! ' in <i>PS1</i> with the history file number of the next command to be typed. Escaping the ' ! ' with another ' ! ' (that is, " ! ! ") shall place the literal character ' ! ' in the prompt. This volume of IEEE Std 1003.1-2001 specifies the effects of the variable only for systems supporting the User Portability Utilities option.
1465		
1466		
1467		
1468		
1469		
1470		
1471		
1472		
1473		
1474	<i>PS2</i>	Each time the user enters a <newline> prior to completing a command line in an interactive shell, the value of this variable shall be subjected to parameter expansion and written to standard error. The default value is "> ". This volume of IEEE Std 1003.1-2001 specifies the effects of the variable only for systems supporting the User Portability Utilities option.
1475		
1476		
1477		
1478		
1479	<i>PS4</i>	When an execution trace (<i>set -x</i>) is being performed in an interactive shell, before each line in the execution trace, the value of this variable shall be subjected to parameter expansion and written to standard error. The default value is "+ ". This volume of IEEE Std 1003.1-2001 specifies the effects of the variable only for systems supporting the User Portability Utilities option.
1480		
1481		
1482		
1483		
1484	<i>PWD</i>	Set by the shell to be an absolute pathname of the current working directory, containing no components of type symbolic link, no components that are dot, and no components that are dot-dot when the shell is initialized. If an application sets or unsets the value of <i>PWD</i> , the behaviors of the <i>cd</i> and <i>pwd</i> utilities are unspecified.
1485		
1486		
1487		
1488		

2.6 Word Expansions

This section describes the various expansions that are performed on words. Not all expansions are performed on every word, as explained in the following sections.

Tilde expansions, parameter expansions, command substitutions, arithmetic expansions, and quote removals that occur within a single word expand to a single field. It is only field splitting or pathname expansion that can create multiple fields from a single word. The single exception to this rule is the expansion of the special parameter '@' within double-quotes, as described in Section 2.5.2 (on page 34).

The order of word expansion shall be as follows:

1. Tilde expansion (see Section 2.6.1 (on page 37)), parameter expansion (see Section 2.6.2 (on page 37)), command substitution (see Section 2.6.3 (on page 40)), and arithmetic expansion (see Section 2.6.4 (on page 41)) shall be performed, beginning to end. See item 5 in Section 2.3 (on page 31).

- 1502 2. Field splitting (see Section 2.6.5 (on page 42)) shall be performed on the portions of the
 1503 fields generated by step 1, unless *IFS* is null.
- 1504 3. Pathname expansion (see Section 2.6.6 (on page 42)) shall be performed, unless *set -f* is in
 1505 effect.
- 1506 4. Quote removal (see Section 2.6.7 (on page 42)) shall always be performed last.

1507 The expansions described in this section shall occur in the same shell environment as that in
 1508 which the command is executed.

1509 If the complete expansion appropriate for a word results in an empty field, that empty field shall
 1510 be deleted from the list of fields that form the completely expanded command, unless the
 1511 original word contained single-quote or double-quote characters.

1512 The '\$' character is used to introduce parameter expansion, command substitution, or
 1513 arithmetic evaluation. If an unquoted '\$' is followed by a character that is either not numeric,
 1514 the name of one of the special parameters (see Section 2.5.2 (on page 34)), a valid first character
 1515 of a variable name, a left curly brace ('{') or a left parenthesis, the result is unspecified.

1516 2.6.1 Tilde Expansion

1517 A "tilde-prefix" consists of an unquoted tilde character at the beginning of a word, followed by
 1518 all of the characters preceding the first unquoted slash in the word, or all the characters in the
 1519 word if there is no slash. In an assignment (see the Base Definitions volume of
 1520 IEEE Std 1003.1-2001, Section 4.21, Variable Assignment), multiple tilde-prefixes can be used: at
 1521 the beginning of the word (that is, following the equal sign of the assignment), following any
 1522 unquoted colon, or both. A tilde-prefix in an assignment is terminated by the first unquoted
 1523 colon or slash. If none of the characters in the tilde-prefix are quoted, the characters in the tilde-
 1524 prefix following the tilde are treated as a possible login name from the user database. A portable
 1525 login name cannot contain characters outside the set given in the description of the *LOGNAME*
 1526 environment variable in the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.3, Other
 1527 Environment Variables. If the login name is null (that is, the tilde-prefix contains only the tilde),
 1528 the tilde-prefix is replaced by the value of the variable *HOME*. If *HOME* is unset, the results are
 1529 unspecified. Otherwise, the tilde-prefix shall be replaced by a pathname of the initial working
 1530 directory associated with the login name obtained using the *getpwnam()* function as defined in
 1531 the System Interfaces volume of IEEE Std 1003.1-2001. If the system does not recognize the login
 1532 name, the results are undefined.

1533 2.6.2 Parameter Expansion

1534 The format for parameter expansion is as follows:

1535 \${*expression*}

1536 where *expression* consists of all characters until the matching '}'. Any '}' escaped by a
 1537 backslash or within a quoted string, and characters in embedded arithmetic expansions,
 1538 command substitutions, and variable expansions, shall not be examined in determining the
 1539 matching '}'.

1540 The simplest form for parameter expansion is:

1541 \${*parameter*}

1542 The value, if any, of *parameter* shall be substituted.

1543 The parameter name or symbol can be enclosed in braces, which are optional except for
 1544 positional parameters with more than one digit or when *parameter* is followed by a character that
 1545 could be interpreted as part of the name. The matching closing brace shall be determined by

		counting brace levels, skipping over enclosed quoted strings, and command substitutions.	
1547		If the parameter name or symbol is not enclosed in braces, the expansion shall use the longest valid name (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.230, Name), whether or not the symbol represented by that name exists.	
1548			
1549			
1550		If a parameter expansion occurs inside double-quotes:	
1551		• Pathname expansion shall not be performed on the results of the expansion.	
1552		• Field splitting shall not be performed on the results of the expansion, with the exception of	
1553		'@'; see Section 2.5.2 (on page 34).	
1554		In addition, a parameter expansion can be modified by using one of the following formats. In	
1555		each case that a value of <i>word</i> is needed (based on the state of <i>parameter</i> , as described below),	
1556		<i>word</i> shall be subjected to tilde expansion, parameter expansion, command substitution, and	
1557		arithmetic expansion. If <i>word</i> is not needed, it shall not be expanded. The '}' character that	
1558		delimits the following parameter expansion modifications shall be determined as described	
1559		previously in this section and in Section 2.2.3 (on page 30). (For example, \${foo-bar}xyz} would	
1560		result in the expansion of foo followed by the string xyz} if foo is set, else the string	
1561		"barxyz}").	
1562	\${parameter:-word}	Use Default Values. If <i>parameter</i> is unset or null, the expansion of <i>word</i> shall be substituted; otherwise, the value of <i>parameter</i> shall be substituted.	
1563			
1564	\${parameter:=word}	Assign Default Values. If <i>parameter</i> is unset or null, the expansion of <i>word</i> shall be assigned to <i>parameter</i> . In all cases, the final value of <i>parameter</i> shall be substituted. Only variables, not positional parameters or special parameters, can be assigned in this way.	
1565			
1566			
1567			
1568	\${parameter?:[word]}	Indicate Error if Null or Unset. If <i>parameter</i> is unset or null, the expansion of <i>word</i> (or a message indicating it is unset if <i>word</i> is omitted) shall be written to standard error and the shell exits with a non-zero exit status. Otherwise, the value of <i>parameter</i> shall be substituted. An interactive shell need not exit.	
1569			
1570			
1571			
1572			
1573	\${parameter:+word}	Use Alternative Value. If <i>parameter</i> is unset or null, null shall be substituted; otherwise, the expansion of <i>word</i> shall be substituted.	
1574			
1575		In the parameter expansions shown previously, use of the colon in the format shall result in a	
1576		test for a parameter that is unset or null; omission of the colon shall result in a test for a	
1577		parameter that is only unset. The following table summarizes the effect of the colon:	
1578			
1579			
	<i>parameter</i> Set and Not Null	<i>parameter</i> Set But Null	<i>parameter</i> Unset
1580	\${parameter:-word}	substitute <i>parameter</i>	substitute <i>word</i>
1581	\${parameter=word}	substitute <i>parameter</i>	substitute <i>word</i>
1582	\${parameter:=word}	substitute <i>parameter</i>	assign <i>word</i>
1583	\${parameter=word}	substitute <i>parameter</i>	substitute <i>null</i>
1584	\${parameter?word}	substitute <i>parameter</i>	error, exit
1585	\${parameter?word}	substitute <i>parameter</i>	substitute <i>null</i>
1586	\${parameter:+word}	substitute <i>word</i>	substitute <i>null</i>
1587	\${parameter+word}	substitute <i>word</i>	substitute <i>null</i>

In all cases shown with "substitute", the expression is replaced with the value shown. In all cases shown with "assign", *parameter* is assigned that value, which also replaces the expression.

1590 \${#parameter} **String Length.** The length in characters of the value of *parameter* shall be
 1591 substituted. If *parameter* is '*' or '@', the result of the expansion is
 1592 unspecified.

1593 The following four varieties of parameter expansion provide for substring processing. In each
 1594 case, pattern matching notation (see Section 2.13 (on page 62)), rather than regular expression
 1595 notation, shall be used to evaluate the patterns. If *parameter* is '*' or '@', the result of the
 1596 expansion is unspecified. Enclosing the full parameter expansion string in double-quotes shall
 1597 not cause the following four varieties of pattern characters to be quoted, whereas quoting
 1598 characters within the braces shall have this effect.

1599 \${parameter%word} **Remove Smallest Suffix Pattern.** The *word* shall be expanded to produce
 1600 a pattern. The parameter expansion shall then result in *parameter*, with the
 1601 smallest portion of the suffix matched by the *pattern* deleted.

1602 \${parameter%%word} **Remove Largest Suffix Pattern.** The *word* shall be expanded to produce a
 1603 pattern. The parameter expansion shall then result in *parameter*, with the
 1604 largest portion of the suffix matched by the *pattern* deleted.

1605 \${parameter#word} **Remove Smallest Prefix Pattern.** The *word* shall be expanded to produce a
 1606 pattern. The parameter expansion shall then result in *parameter*, with the
 1607 smallest portion of the prefix matched by the *pattern* deleted.

1608 \${parameter##word} **Remove Largest Prefix Pattern.** The *word* shall be expanded to produce a
 1609 pattern. The parameter expansion shall then result in *parameter*, with the
 1610 largest portion of the prefix matched by the *pattern* deleted.

1611 Examples

1612 \${parameter:−word}

1613 In this example, *ls* is executed only if *x* is null or unset. (The \${*ls*} command substitution
 1614 notation is explained in Section 2.6.3 (on page 40).)

1615 \$ {x:-\$(ls)}

1616 \${parameter:=word}

```
1617        unset X
1618        echo ${X:=abc}
1619        abc
```

1620 \${parameter?:word}

```
1621        unset posix
1622        echo ${posix:?}
1623        sh: posix: parameter null or not set
```

1624 \${parameter:+word}

```
1625        set a b c
1626        echo ${3:+posix}
1627        posix
```

1628 \${#parameter}

```
1629        HOME=/usr posix
1630        echo ${#HOME}
1631        10
```

1632 \${parameter%word}

```
1633        x=file.c
1634        echo ${x%.c}.o
```

```

1635      file.o
1636      ${parameter%%word}
1637      x=posix/src/std
1638      echo ${x%/*}
1639      posix
1640      ${parameter#word}
1641      x=$HOME/src/cmd
1642      echo ${x#$HOME}
1643      /src/cmd
1644      ${parameter##word}
1645      x=/one/two/three
1646      echo ${x##*/}
1647      three
1648 The double-quoting of patterns is different depending on where the double-quotes are placed:
1649      " ${x##*}"      The asterisk is a pattern character.
1650      ${x##*}" }      The literal asterisk is quoted and not special.

```

1651 2.6.3 Command Substitution

1652 Command substitution allows the output of a command to be substituted in place of the
 1653 command name itself. Command substitution shall occur when the command is enclosed as
 1654 follows:

1655 \$(*command*)

1656 or (backquoted version):

1657 `*command*`

1658 The shell shall expand the command substitution by executing *command* in a subshell
 1659 environment (see Section 2.12 (on page 61)) and replacing the command substitution (the text of
 1660 *command* plus the enclosing "\$()" or backquotes) with the standard output of the command,
 1661 removing sequences of one or more <newline>s at the end of the substitution. Embedded
 1662 <newline>s before the end of the output shall not be removed; however, they may be treated as
 1663 field delimiters and eliminated during field splitting, depending on the value of IFS and quoting
 1664 that is in effect.

1665 Within the backquoted style of command substitution, backslash shall retain its literal meaning,
 1666 except when followed by: '\$', ``', or '\` (dollar sign, backquote, backslash). The search for
 1667 the matching backquote shall be satisfied by the first backquote found without a preceding
 1668 backslash; during this search, if a non-escaped backquote is encountered within a shell
 1669 comment, a here-document, an embedded command substitution of the \$(*command*) form, or a
 1670 quoted string, undefined results occur. A single-quoted or double-quoted string that begins, but
 1671 does not end, within the ``...`` sequence produces undefined results.

1672 With the \$(*command*) form, all characters following the open parenthesis to the matching closing
 1673 parenthesis constitute the *command*. Any valid shell script can be used for *command*, except a
 1674 script consisting solely of redirections which produces unspecified results.

1675 The results of command substitution shall not be processed for further tilde expansion,
 1676 parameter expansion, command substitution, or arithmetic expansion. If a command
 1677 substitution occurs inside double-quotes, it shall not be performed on the results of the
 1678 substitution.

1679 Command substitution can be nested. To specify nesting within the backquoted version, the
 1680 application shall precede the inner backquotes with backslashes, for example:

1681 `\`command\``

1682 If the command substitution consists of a single subshell, such as:

1683 `$((command))`

1684 a conforming application shall separate the `"$()` and `'()'` into two tokens (that is, separate
 1685 them with white space). This is required to avoid any ambiguities with arithmetic expansion.

1686 2.6.4 Arithmetic Expansion

1687 Arithmetic expansion provides a mechanism for evaluating an arithmetic expression and
 1688 substituting its value. The format for arithmetic expansion shall be as follows:

1689 `$((expression))`

1690 The expression shall be treated as if it were in double-quotes, except that a double-quote inside
 1691 the expression is not treated specially. The shell shall expand all tokens in the expression for
 1692 parameter expansion, command substitution, and quote removal.

1693 Next, the shell shall treat this as an arithmetic expression and substitute the value of the
 1694 expression. The arithmetic expression shall be processed according to the rules given in Section
 1695 1.7.2.1 (on page 7), with the following exceptions:

- 1696 • Only signed long integer arithmetic is required.
- 1697 • Only the decimal-constant, octal-constant, and hexadecimal-constant constants specified in
 the ISO C standard, Section 6.4.4.1 are required to be recognized as constants.
- 1699 • The `sizeof()` operator and the prefix and postfix `"++"` and `"--"` operators are not required.
- 1700 • Selection, iteration, and jump statements are not supported.

1701 As an extension, the shell may recognize arithmetic expressions beyond those listed. The shell
 1702 may use a signed integer type with a rank larger than the rank of **signed long**. The shell may use
 1703 a real-floating type instead of **signed long** as long as it does not affect the results in cases where
 1704 there is no overflow. If the expression is invalid, the expansion fails and the shell shall write a
 1705 message to standard error indicating the failure.

1706 Examples

1707 A simple example using arithmetic expansion:

```
1708 # repeat a command 100 times
1709 x=100
1710 while [ $x -gt 0 ]
1711 do
1712     command
1713     x=$((x-1))
1714 done
```

1715 2.6.5 Field Splitting

1716 After parameter expansion (Section 2.6.2 (on page 37)), command substitution (Section 2.6.3 (on
1717 page 40)), and arithmetic expansion (Section 2.6.4 (on page 41)), the shell shall scan the results of
1718 expansions and substitutions that did not occur in double-quotes for field splitting and multiple
1719 fields can result.

1720 The shell shall treat each character of the *IFS* as a delimiter and use the delimiters to split the
1721 results of parameter expansion and command substitution into fields.

- 1722 1. If the value of *IFS* is a <space>, <tab>, and <newline>, or if it is unset, any sequence of
1723 <space>s, <tab>s, or <newline>s at the beginning or end of the input shall be ignored and
1724 any sequence of those characters within the input shall delimit a field. For example, the
1725 input:

1726 <newline><space><tab>foo<tab><tab>bar<space>

1727 yields two fields, **foo** and **bar**.

- 1728 2. If the value of *IFS* is null, no field splitting shall be performed.
- 1729 3. Otherwise, the following rules shall be applied in sequence. The term “*IFS* white space” is
1730 used to mean any sequence (zero or more instances) of white space characters that are in
1731 the *IFS* value (for example, if *IFS* contains <space>/<comma>/<tab>, any sequence of
1732 <space>s and <tab>s is considered *IFS* white space).
- 1733 a. *IFS* white space shall be ignored at the beginning and end of the input.
- 1734 b. Each occurrence in the input of an *IFS* character that is not *IFS* white space, along
1735 with any adjacent *IFS* white space, shall delimit a field, as described previously.
- 1736 c. Non-zero-length *IFS* white space shall delimit a field.

1737 2.6.6 Pathname Expansion

1738 After field splitting, if *set -f* is not in effect, each field in the resulting command line shall be
1739 expanded using the algorithm described in Section 2.13 (on page 62), qualified by the rules in
1740 Section 2.13.3 (on page 63).

1741 2.6.7 Quote Removal

1742 The quote characters: ‘\’, ‘ ’, and ‘ ”’ (backslash, single-quote, double-quote) that were
1743 present in the original word shall be removed unless they have themselves been quoted.

1744 2.7 Redirection

1745 Redirection is used to open and close files for the current shell execution environment (see
1746 Section 2.12 (on page 61)) or for any command. Redirection operators can be used with numbers
1747 representing file descriptors (see the Base Definitions volume of IEEE Std 1003.1-2001, Section
1748 3.165, File Descriptor) as described below.

1749 The overall format used for redirection is:

1750 `[n]redir-op word`

1751 The number *n* is an optional decimal number designating the file descriptor number; the
1752 application shall ensure it is delimited from any preceding text and immediately precede the
1753 redirection operator *redir-op*. If *n* is quoted, the number shall not be recognized as part of the
1754 redirection expression. For example:

1755 `echo \2>a`

1756 writes the character 2 into file a. If any part of *redir-op* is quoted, no redirection expression is
1757 recognized. For example:

1758 `echo 2\>a`

1759 writes the characters 2>a to standard output. The optional number, redirection operator, and
1760 word shall not appear in the arguments provided to the command to be executed (if any).

1761 Open files are represented by decimal numbers starting with zero. The largest possible value is
1762 implementation-defined; however, all implementations shall support at least 0 to 9, inclusive, for
1763 use by the application. These numbers are called “file descriptors”. The values 0, 1, and 2 have
1764 special meaning and conventional uses and are implied by certain redirection operations; they
1765 are referred to as *standard input*, *standard output*, and *standard error*, respectively. Programs
1766 usually take their input from standard input, and write output on standard output. Error
1767 messages are usually written on standard error. The redirection operators can be preceded by
1768 one or more digits (with no intervening <blank>s allowed) to designate the file descriptor
1769 number.

1770 If the redirection operator is "<<" or "<<-", the word that follows the redirection operator shall
1771 be subjected to quote removal; it is unspecified whether any of the other expansions occur. For
1772 the other redirection operators, the word that follows the redirection operator shall be subjected
1773 to tilde expansion, parameter expansion, command substitution, arithmetic expansion, and
1774 quote removal. Pathname expansion shall not be performed on the word by a non-interactive
1775 shell; an interactive shell may perform it, but shall do so only when the expansion would result
1776 in one word.

1777 If more than one redirection operator is specified with a command, the order of evaluation is
1778 from beginning to end.

1779 A failure to open or create a file shall cause a redirection to fail.

1780 **2.7.1 Redirecting Input**

1781 Input redirection shall cause the file whose name results from the expansion of *word* to be
 1782 opened for reading on the designated file descriptor, or standard input if the file descriptor is not
 1783 specified.

1784 The general format for redirecting input is:

1785 [*n*]<*word*

1786 where the optional *n* represents the file descriptor number. If the number is omitted, the
 1787 redirection shall refer to standard input (file descriptor 0).

1788 **2.7.2 Redirecting Output**

1789 The two general formats for redirecting output are:

1790 [*n*]>*word*

1791 [*n*]>|*word*

1792 where the optional *n* represents the file descriptor number. If the number is omitted, the
 1793 redirection shall refer to standard output (file descriptor 1).

1794 Output redirection using the '>' format shall fail if the *noclobber* option is set (see the
 1795 description of *set -C*) and the file named by the expansion of *word* exists and is a regular file.
 1796 Otherwise, redirection using the '>' or ">|" formats shall cause the file whose name results
 1797 from the expansion of *word* to be created and opened for output on the designated file
 1798 descriptor, or standard output if none is specified. If the file does not exist, it shall be created;
 1799 otherwise, it shall be truncated to be an empty file after being opened.

1800 **2.7.3 Appending Redirected Output**

1801 Appended output redirection shall cause the file whose name results from the expansion of
 1802 *word* to be opened for output on the designated file descriptor. The file is opened as if the *open()*
 1803 function as defined in the System Interfaces volume of IEEE Std 1003.1-2001 was called with the
 1804 *O_APPEND* flag. If the file does not exist, it shall be created.

1805 The general format for appending redirected output is as follows:

1806 [*n*]>>*word*

1807 where the optional *n* represents the file descriptor number. If the number is omitted, the
 1808 redirection refers to standard output (file descriptor 1).

1809 **2.7.4 Here-Document**

1810 The redirection operators "<<" and "<<-" both allow redirection of lines contained in a shell
 1811 input file, known as a "here-document", to the input of a command.

1812 The here-document shall be treated as a single word that begins after the next <newline> and
 1813 continues until there is a line containing only the delimiter and a <newline>, with no <blank>s in
 1814 between. Then the next here-document starts, if there is one. The format is as follows:

1815 [*n*]<<*word*
 1816 *here-document*
 1817 *delimiter*

1818 where the optional *n* represents the file descriptor number. If the number is omitted, the here-
 1819 document refers to standard input (file descriptor 0).

1820 If any character in *word* is quoted, the delimiter shall be formed by performing quote removal on
1821 *word*, and the here-document lines shall not be expanded. Otherwise, the delimiter shall be the
1822 *word* itself.

1823 If no characters in *word* are quoted, all lines of the here-document shall be expanded for
1824 parameter expansion, command substitution, and arithmetic expansion. In this case, the
1825 backslash in the input behaves as the backslash inside double-quotes (see Section 2.2.3 (on page
1826 30)). However, the double-quote character ('"') shall not be treated specially within a here-
1827 document, except when the double-quote appears within "\$()", "''", or "\${}".

1828 If the redirection symbol is "<<-", all leading <tab>s shall be stripped from input lines and the
1829 line containing the trailing delimiter. If more than one "<<" or "<<-" operator is specified on a
1830 line, the here-document associated with the first operator shall be supplied first by the
1831 application and shall be read first by the shell.

1832 Examples

1833 An example of a here-document follows:

```
1834 cat <<eof1; cat <<eof2
1835     Hi,
1836     eof1
1837     Helene.
1838     eof2
```

1839 2.7.5 Duplicating an Input File Descriptor

1840 The redirection operator:

```
1841 [n]<&word
```

1842 shall duplicate one input file descriptor from another, or shall close one. If *word* evaluates to one
1843 or more digits, the file descriptor denoted by *n*, or standard input if *n* is not specified, shall be
1844 made to be a copy of the file descriptor denoted by *word*; if the digits in *word* do not represent a
1845 file descriptor already open for input, a redirection error shall result; see Section 2.8.1 (on page
1846 46). If *word* evaluates to '-', file descriptor *n*, or standard input if *n* is not specified, shall be
1847 closed. Attempts to close a file descriptor that is not open shall not constitute an error. If *word*
1848 evaluates to something else, the behavior is unspecified.

1849 2.7.6 Duplicating an Output File Descriptor

1850 The redirection operator:

```
1851 [n]>&word
```

1852 shall duplicate one output file descriptor from another, or shall close one. If *word* evaluates to
1853 one or more digits, the file descriptor denoted by *n*, or standard output if *n* is not specified, shall
1854 be made to be a copy of the file descriptor denoted by *word*; if the digits in *word* do not represent
1855 a file descriptor already open for output, a redirection error shall result; see Section 2.8.1 (on page
1856 46). If *word* evaluates to '-', file descriptor *n*, or standard output if *n* is not specified, is
1857 closed. Attempts to close a file descriptor that is not open shall not constitute an error. If *word*
1858 evaluates to something else, the behavior is unspecified.

1859 **2.7.7 Open File Descriptors for Reading and Writing**

1860 The redirection operator:

1861 `[n]<>word`

1862 shall cause the file whose name is the expansion of *word* to be opened for both reading and
 1863 writing on the file descriptor denoted by *n*, or standard input if *n* is not specified. If the file does
 1864 not exist, it shall be created.

1865 **2.8 Exit Status and Errors**

1866 **2.8.1 Consequences of Shell Errors**

1867 For a non-interactive shell, an error condition encountered by a special built-in (see Section 2.14
 1868 (on page 64)) or other type of utility shall cause the shell to write a diagnostic message to
 1869 standard error and exit as shown in the following table:

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

1878 An expansion error is one that occurs when the shell expansions defined in Section 2.6 (on page
 1879 36) are carried out (for example, "`$(x!y)`", because '`!`' is not a valid operator); an
 1880 implementation may treat these as syntax errors if it is able to detect them during tokenization,
 1881 rather than during expansion.

1882 If any of the errors shown as "shall exit" or "(may) exit" occur in a subshell, the subshell shall
 1883 (respectively may) exit with a non-zero status, but the script containing the subshell shall not
 1884 exit because of the error.

1885 In all of the cases shown in the table, an interactive shell shall write a diagnostic message to
 1886 standard error without exiting.

1887 **2.8.2 Exit Status for Commands**

1888 Each command has an exit status that can influence the behavior of other shell commands. The
 1889 exit status of commands that are not utilities is documented in this section. The exit status of the
 1890 standard utilities is documented in their respective sections.

1891 If a command is not found, the exit status shall be 127. If the command name is found, but it is
 1892 not an executable utility, the exit status shall be 126. Applications that invoke utilities without
 1893 using the shell should use these exit status values to report similar errors.

1894 If a command fails during word expansion or redirection, its exit status shall be greater than
 1895 zero.

1896 Internally, for purposes of deciding whether a command exits with a non-zero exit status, the
 1897 shell shall recognize the entire status value retrieved for the command by the equivalent of the
 1898 `wait()` function `WEXITSTATUS` macro (as defined in the System Interfaces volume of
 1899 IEEE Std 1003.1-2001). When reporting the exit status with the special parameter '?', the shell

shall report the full eight bits of exit status available. The exit status of a command that terminated because it received a signal shall be reported as greater than 128.

2.9 Shell Commands

This section describes the basic structure of shell commands. The following command descriptions each describe a format of the command that is only used to aid the reader in recognizing the command type, and does not formally represent the syntax. Each description discusses the semantics of the command; for a formal definition of the command language, consult Section 2.10 (on page 55).

A *command* is one of the following:

- Simple command (see Section 2.9.1)
- Pipeline (see Section 2.9.2 (on page 49))
- List compound-list (see Section 2.9.3 (on page 50))
- Compound command (see Section 2.9.4 (on page 52))
- Function definition (see Section 2.9.5 (on page 54))

Unless otherwise stated, the exit status of a command shall be that of the last simple command executed by the command. There shall be no limit on the size of any shell command other than that imposed by the underlying system (memory constraints, {ARG_MAX}, and so on).

2.9.1 Simple Commands

A “simple command” is a sequence of optional variable assignments and redirections, in any sequence, optionally followed by words and redirections, terminated by a control operator.

When a given simple command is required to be executed (that is, when any conditional construct such as an AND-OR list or a *case* statement has not bypassed the simple command), the following expansions, assignments, and redirections shall all be performed from the beginning of the command text to the end:

1. The words that are recognized as variable assignments or redirections according to Section 2.10.2 (on page 56) are saved for processing in steps 3 and 4.
2. The words that are not variable assignments or redirections shall be expanded. If any fields remain following their expansion, the first field shall be considered the command name and remaining fields are the arguments for the command.
3. Redirections shall be performed as described in Section 2.7 (on page 43).
4. Each variable assignment shall be expanded for tilde expansion, parameter expansion, command substitution, arithmetic expansion, and quote removal prior to assigning the value.

In the preceding list, the order of steps 3 and 4 may be reversed for the processing of special built-in utilities; see Section 2.14 (on page 64).

If no command name results, variable assignments shall affect the current execution environment. Otherwise, the variable assignments shall be exported for the execution environment of the command and shall not affect the current execution environment (except for special built-ins). If any of the variable assignments attempt to assign a value to a read-only variable, a variable assignment error shall occur. See Section 2.8.1 (on page 46) for the consequences of these errors.

If there is no command name, any redirections shall be performed in a subshell environment; it is unspecified whether this subshell environment is the same one as that used for a command substitution within the command. (To affect the current execution environment, see the `exec` special built-in.) If any of the redirections performed in the current shell execution environment fail, the command shall immediately fail with an exit status greater than zero, and the shell shall write an error message indicating the failure. See Section 2.8.1 (on page 46) for the consequences of these failures on interactive and non-interactive shells.

If there is a command name, execution shall continue as described in Section 2.9.1.1. If there is no command name, but the command contained a command substitution, the command shall complete with the exit status of the last command substitution performed. Otherwise, the command shall complete with a zero exit status.

2.9.1.1 Command Search and Execution

If a simple command results in a command name and an optional list of arguments, the following actions shall be performed:

1. If the command name does not contain any slashes, the first successful step in the following sequence shall occur:
 - a. If the command name matches the name of a special built-in utility, that special built-in utility shall be invoked.
 - b. If the command name matches the name of a function known to this shell, the function shall be invoked as described in Section 2.9.5 (on page 54). If the implementation has provided a standard utility in the form of a function, it shall not be recognized at this point. It shall be invoked in conjunction with the path search in step 1d.
 - c. If the command name matches the name of a utility listed in the following table, that utility shall be invoked.

<i>alias</i>	<i>false</i>	<i>jobs</i>	<i>read</i>	<i>wait</i>
<i>bg</i>	<i>fc</i>	<i>kill</i>	<i>true</i>	
<i>cd</i>	<i>fg</i>	<i>newgrp</i>	<i>umask</i>	
<i>command</i>	<i>getopts</i>	<i>pwd</i>	<i>unalias</i>	

- d. Otherwise, the command shall be searched for using the *PATH* environment variable as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables:

- i. If the search is successful:

- a. If the system has implemented the utility as a regular built-in or as a shell function, it shall be invoked at this point in the path search.
- b. Otherwise, the shell executes the utility in a separate utility environment (see Section 2.12 (on page 61)) with actions equivalent to calling the `execve()` function as defined in the System Interfaces volume of IEEE Std 1003.1-2001 with the *path* argument set to the pathname resulting from the search, *arg0* set to the command name, and the remaining arguments set to the operands, if any.

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

1986 passed to the new shell. If the executable file is not a text file, the shell
1987 may bypass this command execution. In this case, it shall write an error
1988 message, and shall return an exit status of 126.

1989 Once a utility has been searched for and found (either as a result of this specific
1990 search or as part of an unspecified shell start-up activity), an implementation
1991 may remember its location and need not search for the utility again unless the
1992 *PATH* variable has been the subject of an assignment. If the remembered
1993 location fails for a subsequent invocation, the shell shall repeat the search to
1994 find the new location for the utility, if any.

1995 ii. If the search is unsuccessful, the command shall fail with an exit status of 127
1996 and the shell shall write an error message.

1997 2. If the command name contains at least one slash, the shell shall execute the utility in a
1998 separate utility environment with actions equivalent to calling the *execve()* function
1999 defined in the System Interfaces volume of IEEE Std 1003.1-2001 with the *path* and *arg0*
2000 arguments set to the command name, and the remaining arguments set to the operands, if
2001 any.

2002 If the *execve()* function fails due to an error equivalent to the [ENOEXEC] error, the shell
2003 shall execute a command equivalent to having a shell invoked with the command name as
2004 its first operand, with any remaining arguments passed to the new shell. If the executable
2005 file is not a text file, the shell may bypass this command execution. In this case, it shall
2006 write an error message and shall return an exit status of 126.

2007 2.9.2 Pipelines

2008 A *pipeline* is a sequence of one or more commands separated by the control operator '|'. The
2009 standard output of all but the last command shall be connected to the standard input of the next
2010 command.

2011 The format for a pipeline is:

2012 [!] *command1* [| *command2* ...]

2013 The standard output of *command1* shall be connected to the standard input of *command2*. The
2014 standard input, standard output, or both of a command shall be considered to be assigned by the
2015 pipeline before any redirection specified by redirection operators that are part of the command
2016 (see Section 2.7 (on page 43)).

2017 If the pipeline is not in the background (see Section 2.9.3.1 (on page 50)), the shell shall wait for
2018 the last command specified in the pipeline to complete, and may also wait for all commands to
2019 complete.

2020 Exit Status

2021 If the reserved word ! does not precede the pipeline, the exit status shall be the exit status of the
2022 last command specified in the pipeline. Otherwise, the exit status shall be the logical NOT of the
2023 exit status of the last command. That is, if the last command returns zero, the exit status shall be
2024 1; if the last command returns greater than zero, the exit status shall be zero.

2025 2.9.3 Lists

2026 An *AND-OR list* is a sequence of one or more pipelines separated by the operators "`&&`" and
2027 "`||`".

2028 A *list* is a sequence of one or more AND-OR lists separated by the operators '`;`' and '`&`' and
2029 optionally terminated by '`;`', '`&`', or `<newline>`.

2030 The operators "`&&`" and "`||`" shall have equal precedence and shall be evaluated with left
2031 associativity. For example, both of the following commands write solely **bar** to standard output:

```
2032     false && echo foo || echo bar
2033     true || echo foo && echo bar
```

2034 A '`;`' or `<newline>` terminator shall cause the preceding AND-OR list to be executed
2035 sequentially; an '`&`' shall cause asynchronous execution of the preceding AND-OR list.

2036 The term "compound-list" is derived from the grammar in Section 2.10 (on page 55); it is
2037 equivalent to a sequence of *lists*, separated by `<newline>`s, that can be preceded or followed by
2038 an arbitrary number of `<newline>`s.

2039 Examples

2040 The following is an example that illustrates `<newline>`s in compound-lists:

```
2041     while
2042         # a couple of <newline>s
2043
2044         # a list
2045         date && who || ls; cat file
2046         # a couple of <newline>s
2047
2048         # another list
2049         wc file > output & true
2050
2051     do
2052         # 2 lists
2053         ls
2054         cat file
2055     done
```

2053 2.9.3.1 Asynchronous Lists

2054 If a command is terminated by the control operator ampersand ('`&`'), the shell shall execute the
2055 command asynchronously in a subshell. This means that the shell shall not wait for the
2056 command to finish before executing the next command.

2057 The format for running a command in the background is:

```
2058     command1 & [command2 & ... ]
```

2059 The standard input for an asynchronous list, before any explicit redirections are performed, shall
2060 be considered to be assigned to a file that has the same properties as `/dev/null`. If it is an
2061 interactive shell, this need not happen. In all cases, explicit redirection of standard input shall
2062 override this activity.

2063 When an element of an asynchronous list (the portion of the list ended by an ampersand, such as
2064 `command1`, above) is started by the shell, the process ID of the last command in the asynchronous
2065 list element shall become known in the current shell execution environment; see Section 2.12 (on
2066 page 61). This process ID shall remain known until:

- 2067 1. The command terminates and the application waits for the process ID.
2068 2. Another asynchronous list invoked before "\$!" (corresponding to the previous
2069 asynchronous list) is expanded in the current execution environment.

2070 The implementation need not retain more than the {CHILD_MAX} most recent entries in its list
2071 of known process IDs in the current shell execution environment.

2072 **Exit Status**

2073 The exit status of an asynchronous list shall be zero.

2074 **2.9.3.2 Sequential Lists**

2075 Commands that are separated by a semicolon (';') shall be executed sequentially.

2076 The format for executing commands sequentially shall be:

2077 *command1* [*i* *command2*] ...

2078 Each command shall be expanded and executed in the order specified.

2079 **Exit Status**

2080 The exit status of a sequential list shall be the exit status of the last command in the list.

2081 **2.9.3.3 AND Lists**

2082 The control operator "&&" denotes an AND list. The format shall be:

2083 *command1* [*&&* *command2*] ...

2084 First *command1* shall be executed. If its exit status is zero, *command2* shall be executed, and so on,
2085 until a command has a non-zero exit status or there are no more commands left to execute. The
2086 commands are expanded only if they are executed.

2087 **Exit Status**

2088 The exit status of an AND list shall be the exit status of the last command that is executed in the
2089 list.

2090 **2.9.3.4 OR Lists**

2091 The control operator "||" denotes an OR List. The format shall be:

2092 *command1* [*||* *command2*] ...

2093 First, *command1* shall be executed. If its exit status is non-zero, *command2* shall be executed, and
2094 so on, until a command has a zero exit status or there are no more commands left to execute.

2095 **Exit Status**

2096 The exit status of an OR list shall be the exit status of the last command that is executed in the
2097 list.

2098 **2.9.4 Compound Commands**

2099 The shell has several programming constructs that are “compound commands”, which provide
 2100 control flow for commands. Each of these compound commands has a reserved word or control
 2101 operator at the beginning, and a corresponding terminator reserved word or operator at the end.
 2102 In addition, each can be followed by redirections on the same line as the terminator. Each
 2103 redirection shall apply to all the commands within the compound command that do not
 2104 explicitly override that redirection.

2105 **2.9.4.1 Grouping Commands**

2106 The format for grouping commands is as follows:

- 2107 (*compound-list*) Execute *compound-list* in a subshell environment; see Section 2.12 (on page
 2108 61). Variable assignments and built-in commands that affect the
 2109 environment shall not remain in effect after the list finishes.
- 2110 { *compound-list*; } Execute *compound-list* in the current process environment. The semicolon
 2111 shown here is an example of a control operator delimiting the } reserved
 2112 word. Other delimiters are possible, as shown in Section 2.10 (on page
 2113 55); a <newline> is frequently used.

2114 **Exit Status**

2115 The exit status of a grouping command shall be the exit status of *compound-list*.

2116 **2.9.4.2 The for Loop**

2117 The **for** loop shall execute a sequence of commands for each member in a list of *items*. The **for**
 2118 loop requires that the reserved words **do** and **done** be used to delimit the sequence of
 2119 commands.

2120 The format for the **for** loop is as follows:

```
2121   for name [ in [word ... ] ]
2122     do
2123       compound-list
2124     done
```

2125 First, the list of words following **in** shall be expanded to generate a list of items. Then, the
 2126 variable *name* shall be set to each item, in turn, and the *compound-list* executed each time. If no
 2127 items result from the expansion, the *compound-list* shall not be executed. Omitting:

2128 *in word* ...

2129 shall be equivalent to:

2130 *in "\$@"*

2131 **Exit Status**

2132 The exit status of a **for** command shall be the exit status of the last command that executes. If
 2133 there are no items, the exit status shall be zero.

2134 2.9.4.3 Case Conditional Construct

The conditional construct **case** shall execute the *compound-list* corresponding to the first one of several *patterns* (see Section 2.13 (on page 62)) that is matched by the string resulting from the tilde expansion, parameter expansion, command substitution, arithmetic expansion, and quote removal of the given word. The reserved word **in** shall denote the beginning of the patterns to be matched. Multiple patterns with the same *compound-list* shall be delimited by the ' | ' symbol. The control operator ')' terminates a list of patterns corresponding to a given action. The *compound-list* for each list of patterns, with the possible exception of the last, shall be terminated with "; ; ". The **case** construct terminates with the reserved word **esac** (**case** reversed).

The format for the **case** construct is as follows:

```
2144 case word in
2145     [( ]pattern1) compound-list;;
2146     [[( ]pattern[ | pattern] ...) compound-list;;] ...
2147     [[( ]pattern[ | pattern] ...) compound-list]
2148     esac
```

The " ; ; " is optional for the last *compound-list*.

In order from the beginning to the end of the **case** statement, each *pattern* that labels a *compound-list* shall be subjected to tilde expansion, parameter expansion, command substitution, and arithmetic expansion, and the result of these expansions shall be compared against the expansion of *word*, according to the rules described in Section 2.13 (on page 62) (which also describes the effect of quoting parts of the pattern). After the first match, no more patterns shall be expanded, and the *compound-list* shall be executed. The order of expansion and comparison of multiple *patterns* that label a *compound-list* statement is unspecified.

2157 Exit Status

The exit status of **case** shall be zero if no patterns are matched. Otherwise, the exit status shall be the exit status of the last command executed in the *compound-list*.

2160 2.9.4.4 The if Conditional Construct

The **if** command shall execute a *compound-list* and use its exit status to determine whether to execute another *compound-list*.

The format for the **if** construct is as follows:

```
2164 if compound-list
2165     then
2166         compound-list
2167     [elif compound-list
2168     then
2169         compound-list] ...
2170     [else
2171         compound-list]
2172     fi
```

The **if** *compound-list* shall be executed; if its exit status is zero, the **then** *compound-list* shall be executed and the command shall complete. Otherwise, each **elif** *compound-list* shall be executed, in turn, and if its exit status is zero, the **then** *compound-list* shall be executed and the command shall complete. Otherwise, the **else** *compound-list* shall be executed.

- 2177 **Exit Status**
- 2178 The exit status of the **if** command shall be the exit status of the **then** or **else compound-list** that
2179 was executed, or zero, if none was executed.
- 2180 **2.9.4.5 The while Loop**
- 2181 The **while** loop shall continuously execute one *compound-list* as long as another *compound-list* has
2182 a zero exit status.
- 2183 The format of the **while** loop is as follows:
- ```
2184 while compound-list-1
2185 do
2186 compound-list-2
2187 done
```
- 2188         The *compound-list-1* shall be executed, and if it has a non-zero exit status, the **while** command  
2189         shall complete. Otherwise, the *compound-list-2* shall be executed, and the process shall repeat.
- 2190            **Exit Status**
- 2191         The exit status of the **while** loop shall be the exit status of the last *compound-list-2* executed, or  
2192         zero if none was executed.
- 2193     **2.9.4.6 The until Loop**
- 2194         The **until** loop shall continuously execute one *compound-list* as long as another *compound-list* has  
2195         a non-zero exit status.
- 2196         The format of the **until** loop is as follows:
- ```
2197            until compound-list-1
2198            do
2199                compound-list-2
2200            done
```
- 2201 The *compound-list-1* shall be executed, and if it has a zero exit status, the **until** command
2202 completes. Otherwise, the *compound-list-2* shall be executed, and the process repeats.
- 2203 **Exit Status**
- 2204 The exit status of the **until** loop shall be the exit status of the last *compound-list-2* executed, or
2205 zero if none was executed.
- 2206 **2.9.5 Function Definition Command**
- 2207 A function is a user-defined name that is used as a simple command to call a compound
2208 command with new positional parameters. A function is defined with a “function definition
2209 command”.
- 2210 The format of a function definition command is as follows:
- ```
2211 fname () compound-command[io-redirect . . .]
```
- 2212         The function is named *fname*; the application shall ensure that it is a name (see the Base  
2213         Definitions volume of IEEE Std 1003.1-2001, Section 3.230, Name). An implementation may  
2214         allow other characters in a function name as an extension. The implementation shall maintain  
2215         separate name spaces for functions and variables.

2216 The argument *compound-command* represents a compound command, as described in Section  
2217 2.9.4 (on page 52).

2218 When the function is declared, none of the expansions in Section 2.6 (on page 36) shall be  
2219 performed on the text in *compound-command* or *io-redirect*; all expansions shall be performed as  
2220 normal each time the function is called. Similarly, the optional *io-redirect* redirections and any  
2221 variable assignments within *compound-command* shall be performed during the execution of the  
2222 function itself, not the function definition. See Section 2.8.1 (on page 46) for the consequences of  
2223 failures of these operations on interactive and non-interactive shells.

2224 When a function is executed, it shall have the syntax-error and variable-assignment properties  
2225 described for special built-in utilities in the enumerated list at the beginning of Section 2.14 (on  
2226 page 64).

2227 The *compound-command* shall be executed whenever the function name is specified as the name  
2228 of a simple command (see Section 2.9.1.1 (on page 48)). The operands to the command  
2229 temporarily shall become the positional parameters during the execution of the *compound-  
2230 command*; the special parameter '#' also shall be changed to reflect the number of operands. The  
2231 special parameter 0 shall be unchanged. When the function completes, the values of the  
2232 positional parameters and the special parameter '#' shall be restored to the values they had  
2233 before the function was executed. If the special built-in *return* is executed in the *compound-  
2234 command*, the function completes and execution shall resume with the next command after the  
2235 function call.

### 2236 **Exit Status**

2237 The exit status of a function definition shall be zero if the function was declared successfully;  
2238 otherwise, it shall be greater than zero. The exit status of a function invocation shall be the exit  
2239 status of the last command executed by the function.

## 2240 **2.10 Shell Grammar**

2241 The following grammar defines the Shell Command Language. This formal syntax shall take  
2242 precedence over the preceding text syntax description.

### 2243 **2.10.1 Shell Grammar Lexical Conventions**

2244 The input language to the shell must be first recognized at the character level. The resulting  
2245 tokens shall be classified by their immediate context according to the following rules (applied in  
2246 order). These rules shall be used to determine what a "token" is that is subject to parsing at the  
2247 token level. The rules for token recognition in Section 2.3 (on page 31) shall apply.

- 2248 1. A <newline> shall be returned as the token identifier **NEWLINE**.
- 2249 2. If the token is an operator, the token identifier for that operator shall result.
- 2250 3. If the string consists solely of digits and the delimiter character is one of '<' or '>', the  
2251 token identifier **IO\_NUMBER** shall be returned.
- 2252 4. Otherwise, the token identifier **TOKEN** results.

2253 Further distinction on **TOKEN** is context-dependent. It may be that the same **TOKEN** yields  
2254 **WORD**, a **NAME**, an **ASSIGNMENT**, or one of the reserved words below, dependent upon the  
2255 context. Some of the productions in the grammar below are annotated with a rule number from  
2256 the following list. When a **TOKEN** is seen where one of those annotated productions could be  
2257 used to reduce the symbol, the applicable rule shall be applied to convert the token identifier

2258 type of the **TOKEN** to a token identifier acceptable at that point in the grammar. The reduction  
 2259 shall then proceed based upon the token identifier type yielded by the rule applied. When more  
 2260 than one rule applies, the highest numbered rule shall apply (which in turn may refer to another  
 2261 rule). (Note that except in rule 7, the presence of an '=' in the token has no effect.)

2262 The **WORD** tokens shall have the word expansion rules applied to them immediately before the  
 2263 associated command is executed, not at the time the command is parsed.

## 2264 2.10.2 Shell Grammar Rules

### 2265 1. [Command Name]

2266 When the **TOKEN** is exactly a reserved word, the token identifier for that reserved word  
 2267 shall result. Otherwise, the token **WORD** shall be returned. Also, if the parser is in any  
 2268 state where only a reserved word could be the next correct token, proceed as above.

2269 **Note:** Because at this point quote marks are retained in the token, quoted strings cannot be  
 2270 recognized as reserved words. This rule also implies that reserved words are not  
 2271 recognized except in certain positions in the input, such as after a <newline> or  
 2272 semicolon; the grammar presumes that if the reserved word is intended, it is properly  
 2273 delimited by the user, and does not attempt to reflect that requirement directly. Also  
 2274 note that line joining is done before tokenization, as described in Section 2.2.1 (on page  
 2275 30), so escaped <newline>s are already removed at this point.

2276 Rule 1 is not directly referenced in the grammar, but is referred to by other rules, or applies  
 2277 globally.

### 2278 2. [Redirection to or from filename]

2279 The expansions specified in Section 2.7 (on page 43) shall occur. As specified there, exactly  
 2280 one field can result (or the result is unspecified), and there are additional requirements on  
 2281 pathname expansion.

### 2282 3. [Redirection from here-document]

2283 Quote removal shall be applied to the word to determine the delimiter that is used to find  
 2284 the end of the here-document that begins after the next <newline>.

### 2285 4. [Case statement termination]

2286 When the **TOKEN** is exactly the reserved word **esac**, the token identifier for **esac** shall  
 2287 result. Otherwise, the token **WORD** shall be returned.

### 2288 5. [NAME in for]

2289 When the **TOKEN** meets the requirements for a name (see the Base Definitions volume of  
 2290 IEEE Std 1003.1-2001, Section 3.230, Name), the token identifier **NAME** shall result.  
 2291 Otherwise, the token **WORD** shall be returned.

### 2292 6. [Third word of for and case]

#### 2293 a. [case only]

2294 When the **TOKEN** is exactly the reserved word **in**, the token identifier for **in** shall  
 2295 result. Otherwise, the token **WORD** shall be returned.

#### 2296 b. [for only]

2297 When the **TOKEN** is exactly the reserved word **in** or **do**, the token identifier for **in** or  
 2298 **do** shall result, respectively. Otherwise, the token **WORD** shall be returned.

2299 (For a. and b.: As indicated in the grammar, a *linebreak* precedes the tokens **in** and **do**. If  
 2300 <newline>s are present at the indicated location, it is the token after them that is treated in  
 2301 this fashion.)

2302 7. [Assignment preceding command name]

2303 a. [When the first word]

2304 If the **TOKEN** does not contain the character '=' , rule 1 is applied. Otherwise, 7b  
 2305 shall be applied.

2306 b. [Not the first word]

2307 If the **TOKEN** contains the equal sign character:

- 2308 — If it begins with '=', the token **WORD** shall be returned.
- 2309 — If all the characters preceding '=' form a valid name (see the Base Definitions  
 2310 volume of IEEE Std 1003.1-2001, Section 3.230, Name), the token  
 2311 **ASSIGNMENT\_WORD** shall be returned. (Quoted characters cannot participate  
 2312 in forming a valid name.)
- 2313 — Otherwise, it is unspecified whether it is **ASSIGNMENT\_WORD** or **WORD** that  
 2314 is returned.

2315 Assignment to the **NAME** shall occur as specified in Section 2.9.1 (on page 47).

2316 8. [NAME in function]

2317 When the **TOKEN** is exactly a reserved word, the token identifier for that reserved word  
 2318 shall result. Otherwise, when the **TOKEN** meets the requirements for a name, the token  
 2319 identifier **NAME** shall result. Otherwise, rule 7 applies.

2320 9. [Body of function]

2321 Word expansion and assignment shall never occur, even when required by the rules above,  
 2322 when this rule is being parsed. Each **TOKEN** that might either be expanded or have  
 2323 assignment applied to it shall instead be returned as a single **WORD** consisting only of  
 2324 characters that are exactly the token described in Section 2.3 (on page 31).

```
2325 /* -----
2326 The grammar symbols
2327 ----- */
2328 %token WORD
2329 %token ASSIGNMENT_WORD
2330 %token NAME
2331 %token NEWLINE
2332 %token IO_NUMBER
2333 /* The following are the operators mentioned above. */
2334 %token AND_IF OR_IF DSEMI
2335 /* '&&' '||' ';' */
2336 %token DLESS DGREAT LESSAND GREATAND LESSGREAT DLESSDASH
2337 /* '<<' '>>' '<&' '>&' '<>' '<<-'
2338 %token CLOBBER
2339 /* '>|' */
```

```

2340 /* The following are the reserved words. */
2341 %token If Then Else Elif Fi Do Done
2342 /* 'if' 'then' 'else' 'elif' 'fi' 'do' 'done' */
2343 %token Case Esac While Until For
2344 /* 'case' 'esac' 'while' 'until' 'for' */
2345 /* These are reserved words, not operator tokens, and are
2346 recognized when reserved words are recognized. */
2347 %token Lbrace Rbrace Bang
2348 /* '{' '}' '!' */
2349 %token In
2350 /* 'in' */
2351 /* -----
2352 The Grammar
2353 ----- */
2354 %start complete_command
2355 %%
2356 complete_command : list separator
2357 | list
2358 ;
2359 list : list separator_op and_or
2360 | and_or
2361 ;
2362 and_or : pipeline
2363 | and_or AND_IF linebreak pipeline
2364 | and_or OR_IF linebreak pipeline
2365 ;
2366 pipeline : pipe_sequence
2367 | Bang pipe_sequence
2368 ;
2369 pipe_sequence : command
2370 | pipe_sequence '|'
2371 | linebreak command
2372 command : simple_command
2373 | compound_command
2374 | compound_command redirect_list
2375 | function_definition
2376 ;
2377 compound_command : brace_group
2378 | subshell
2379 | for_clause
2380 | case_clause
2381 | if_clause
2382 | while_clause
2383 | until_clause
2384 ;
2385 subshell : '(' compound_list ')'
2386 ;
2387 compound_list : term
2388 | newline_list term

```

```

2389 | term separator
2390 | newline_list term separator
2391 ;
2392 term : term separator and_or
2393 | and_or
2394 ;
2395 for_clause : For name linebreak do_group
2396 | For name linebreak in sequential_sep do_group
2397 | For name linebreak in wordlist sequential_sep do_group
2398 ;
2399 name : NAME /* Apply rule 5 */
2400 ;
2401 in : In /* Apply rule 6 */
2402 ;
2403 wordlist : wordlist WORD
2404 | WORD
2405 ;
2406 case_clause : Case WORD linebreak in linebreak case_list Esac
2407 | Case WORD linebreak in linebreak case_list_ns Esac
2408 | Case WORD linebreak in linebreak Esac
2409 ;
2410 case_list_ns : case_list case_item_ns
2411 | case_item_ns
2412 ;
2413 case_list : case_list case_item
2414 | case_item
2415 ;
2416 case_item_ns : pattern ')' linebreak
2417 | pattern ')' compound_list linebreak
2418 | '(' pattern ')' linebreak
2419 | '(' pattern ')' compound_list linebreak
2420 ;
2421 case_item : pattern ')' linebreak DSEMI linebreak
2422 | pattern ')' compound_list DSEMI linebreak
2423 | '(' pattern ')' linebreak DSEMI linebreak
2424 | '(' pattern ')' compound_list DSEMI linebreak
2425 ;
2426 pattern : WORD /* Apply rule 4 */
2427 | pattern '|' WORD /* Do not apply rule 4 */
2428 ;
2429 if_clause : If compound_list Then compound_list else_part Fi
2430 | If compound_list Then compound_list Fi
2431 ;
2432 else_part : Elif compound_list Then else_part
2433 | Else compound_list
2434 ;
2435 while_clause : While compound_list do_group
2436 ;
2437 until_clause : Until compound_list do_group
2438 ;
2439 function_definition : fname '(' ')' linebreak function_body
2440 ;

```

```

2441 function_body : compound_command /* Apply rule 9 */
2442 | compound_command redirect_list /* Apply rule 9 */
2443 ;
2444 fname : NAME /* Apply rule 8 */
2445 ;
2446 brace_group : Lbrace compound_list Rbrace
2447 ;
2448 do_group : Do compound_list Done /* Apply rule 6 */
2449 ;
2450 simple_command : cmd_prefix cmd_word cmd_suffix
2451 | cmd_prefix cmd_word
2452 | cmd_prefix
2453 | cmd_name cmd_suffix
2454 | cmd_name
2455 ;
2456 cmd_name : WORD /* Apply rule 7a */
2457 ;
2458 cmd_word : WORD /* Apply rule 7b */
2459 ;
2460 cmd_prefix : io_redirect
2461 | cmd_prefix io_redirect
2462 | ASSIGNMENT_WORD
2463 | cmd_prefix ASSIGNMENT_WORD
2464 ;
2465 cmd_suffix : io_redirect
2466 | cmd_suffix io_redirect
2467 | WORD
2468 | cmd_suffix WORD
2469 ;
2470 redirect_list : io_redirect
2471 | redirect_list io_redirect
2472 ;
2473 io_redirect : io_file
2474 | IO_NUMBER io_file
2475 | io_here
2476 | IO_NUMBER io_here
2477 ;
2478 io_file : '<' filename
2479 | LESSAND filename
2480 | '>' filename
2481 | GREATAND filename
2482 | DGREAT filename
2483 | LESSGREAT filename
2484 | CLOBBER filename
2485 ;
2486 filename : WORD /* Apply rule 2 */
2487 ;
2488 io_here : DLESS here_end
2489 | DLESSDASH here_end
2490 ;
2491 here_end : WORD /* Apply rule 3 */
2492 ;

```

```

2493 newline_list : NEWLINE
2494 | newline_list NEWLINE
2495 ;
2496 linebreak : newline_list
2497 | /* empty */
2498 ;
2499 separator_op : '&'
2500 | ';'
2501 ;
2502 separator : separator_op linebreak
2503 | newline_list
2504 ;
2505 sequential_sep : ';' linebreak
2506 | newline_list
2507 ;

```

## 2508 2.11 Signals and Error Handling

2509 When a command is in an asynchronous list, the shell shall prevent SIGQUIT and SIGINT  
 2510 signals from the keyboard from interrupting the command. Otherwise, signals shall have the  
 2511 values inherited by the shell from its parent (see also the *trap* special built-in).

2512 When a signal for which a trap has been set is received while the shell is waiting for the  
 2513 completion of a utility executing a foreground command, the trap associated with that signal  
 2514 shall not be executed until after the foreground command has completed. When the shell is  
 2515 waiting, by means of the *wait* utility, for asynchronous commands to complete, the reception of a  
 2516 signal for which a trap has been set shall cause the *wait* utility to return immediately with an exit  
 2517 status >128, immediately after which the trap associated with that signal shall be taken.

2518 If multiple signals are pending for the shell for which there are associated trap actions, the order  
 2519 of execution of trap actions is unspecified.

## 2520 2.12 Shell Execution Environment

2521 A shell execution environment consists of the following:

- 2522 • Open files inherited upon invocation of the shell, plus open files controlled by *exec*
- 2523 • Working directory as set by *cd*
- 2524 • File creation mask set by *umask*
- 2525 • Current traps set by *trap*
- 2526 • Shell parameters that are set by variable assignment (see the *set* special built-in) or from the  
 2527 System Interfaces volume of IEEE Std 1003.1-2001 environment inherited by the shell when it  
 2528 begins (see the *export* special built-in)
- 2529 • Shell functions; see Section 2.9.5 (on page 54)
- 2530 • Options turned on at invocation or by *set*
- 2531 • Process IDs of the last commands in asynchronous lists known to this shell environment; see  
 2532 Section 2.9.3.1 (on page 50)

- 2533           • Shell aliases; see Section 2.3.1 (on page 32)
- 2534       Utilities other than the special built-ins (see Section 2.14 (on page 64)) shall be invoked in a  
2535       separate environment that consists of the following. The initial value of these objects shall be the  
2536       same as that for the parent shell, except as noted below.
- 2537           • Open files inherited on invocation of the shell, open files controlled by the *exec* special built-  
2538           in plus any modifications, and additions specified by any redirections to the utility
- 2539           • Current working directory
- 2540           • File creation mask
- 2541           • If the utility is a shell script, traps caught by the shell shall be set to the default values and  
2542           traps ignored by the shell shall be set to be ignored by the utility; if the utility is not a shell  
2543           script, the trap actions (default or ignore) shall be mapped into the appropriate signal  
2544           handling actions for the utility
- 2545           • Variables with the *export* attribute, along with those explicitly exported for the duration of the  
2546           command, shall be passed to the utility environment variables

2547       The environment of the shell process shall not be changed by the utility unless explicitly  
2548       specified by the utility description (for example, *cd* and *umask*).

2549       A subshell environment shall be created as a duplicate of the shell environment, except that  
2550       signal traps set by that shell environment shall be set to the default values. Changes made to the  
2551       subshell environment shall not affect the shell environment. Command substitution, commands  
2552       that are grouped with parentheses, and asynchronous lists shall be executed in a subshell  
2553       environment. Additionally, each command of a multi-command pipeline is in a subshell  
2554       environment; as an extension, however, any or all commands in a pipeline may be executed in  
2555       the current environment. All other commands shall be executed in the current shell  
2556       environment.

## 2557 2.13 Pattern Matching Notation

2558       The pattern matching notation described in this section is used to specify patterns for matching  
2559       strings in the shell. Historically, pattern matching notation is related to, but slightly different  
2560       from, the regular expression notation described in the Base Definitions volume of  
2561       IEEE Std 1003.1-2001, Chapter 9, Regular Expressions. For this reason, the description of the  
2562       rules for this pattern matching notation are based on the description of regular expression  
2563       notation, modified to include backslash escape processing.

### 2564 2.13.1 Patterns Matching a Single Character

2565       The following patterns matching a single character shall match a single character: ordinary  
2566       characters, special pattern characters, and pattern bracket expressions. The pattern bracket  
2567       expression also shall match a single collating element. A backslash character shall escape the  
2568       following character. The escaping backslash shall be discarded.

2569       An ordinary character is a pattern that shall match itself. It can be any character in the supported  
2570       character set except for NUL, those special shell characters in Section 2.2 (on page 30) that  
2571       require quoting, and the following three special pattern characters. Matching shall be based on  
2572       the bit pattern used for encoding the character, not on the graphic representation of the  
2573       character. If any character (ordinary, shell special, or pattern special) is quoted, that pattern shall  
2574       match the character itself. The shell special characters always require quoting.

When unquoted and outside a bracket expression, the following three characters shall have special meaning in the specification of patterns:

- ? A question-mark is a pattern that shall match any character.
- \* An asterisk is a pattern that shall match multiple characters, as described in Section 2.13.2.
- [ The open bracket shall introduce a pattern bracket expression.

The description of basic regular expression bracket expressions in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3.5, RE Bracket Expression shall also apply to the pattern bracket expression, except that the exclamation mark character ('!') shall replace the circumflex character ('^') in its role in a “non-matching list” in the regular expression notation. A bracket expression starting with an unquoted circumflex character produces unspecified results.

When pattern matching is used where shell quote removal is not performed (such as in the argument to the *find -name* primary when *find* is being called using one of the *exec* functions as defined in the System Interfaces volume of IEEE Std 1003.1-2001, or in the *pattern* argument to the *fnmatch()* function), special characters can be escaped to remove their special meaning by preceding them with a backslash character. This escaping backslash is discarded. The sequence "\\" represents one literal backslash. All of the requirements and effects of quoting on ordinary, shell special, and special pattern characters shall apply to escaping in this context.

### 2.13.2 Patterns Matching Multiple Characters

The following rules are used to construct patterns matching multiple characters from patterns matching a single character:

1. The asterisk (' \* ') is a pattern that shall match any string, including the null string.
2. The concatenation of patterns matching a single character is a valid pattern that shall match the concatenation of the single characters or collating elements matched by each of the concatenated patterns.
3. The concatenation of one or more patterns matching a single character with one or more asterisks is a valid pattern. In such patterns, each asterisk shall match a string of zero or more characters, matching the greatest possible number of characters that still allows the remainder of the pattern to match the string.

### 2.13.3 Patterns Used for Filename Expansion

The rules described so far in Section 2.13.1 (on page 62) and Section 2.13.2 are qualified by the following rules that apply when pattern matching notation is used for filename expansion:

1. The slash character in a pathname shall be explicitly matched by using one or more slashes in the pattern; it shall neither be matched by the asterisk or question-mark special characters nor by a bracket expression. Slashes in the pattern shall be identified before bracket expressions; thus, a slash cannot be included in a pattern bracket expression used for filename expansion. If a slash character is found following an unescaped open square bracket character before a corresponding closing square bracket is found, the open bracket shall be treated as an ordinary character. For example, the pattern "a[b/c]d" does not match such pathnames as **abd** or **a/d**. It only matches a pathname of literally **a[b/c]d**.
2. If a filename begins with a period (' . '), the period shall be explicitly matched by using a period as the first character of the pattern or immediately following a slash character. The leading period shall not be matched by:

- 2618           • The asterisk or question-mark special characters  
2619           • A bracket expression containing a non-matching list, such as "[ !a ]", a range  
2620           expression, such as "[ %–0 ]", or a character class expression, such as "[ [ :punct: ] ]"  
2621           It is unspecified whether an explicit period in a bracket expression matching list, such as  
2622           " [ .abc ] ", can match a leading period in a filename.  
2623       3. Specified patterns shall be matched against existing filenames and pathnames, as  
2624           appropriate. Each component that contains a pattern character shall require read  
2625           permission in the directory containing that component. Any component, except the last,  
2626           that does not contain a pattern character shall require search permission. For example,  
2627           given the pattern:  
2628           

```
/foo/bar/x*/bam
```

  
2629           search permission is needed for directories / and **foo**, search and read permissions are  
2630           needed for directory **bar**, and search permission is needed for each **x\*** directory. If the  
2631           pattern matches any existing filenames or pathnames, the pattern shall be replaced with  
2632           those filenames and pathnames, sorted according to the collating sequence in effect in the  
2633           current locale. If the pattern contains an invalid bracket expression or does not match any  
2634           existing filenames or pathnames, the pattern string shall be left unchanged.

## 2635 2.14 Special Built-In Utilities

2636 The following "special built-in" utilities shall be supported in the shell command language. The  
2637 output of each command, if any, shall be written to standard output, subject to the normal  
2638 redirection and piping possible with all commands.

2639 The term "built-in" implies that the shell can execute the utility directly and does not need to  
2640 search for it. An implementation may choose to make any utility a built-in; however, the special  
2641 built-in utilities described here differ from regular built-in utilities in two respects:

- 2642     1. A syntax error in a special built-in utility may cause a shell executing that utility to abort,  
2643       while a syntax error in a regular built-in utility shall not cause a shell executing that utility  
2644       to abort. (See Section 2.8.1 (on page 46) for the consequences of errors on interactive and  
2645       non-interactive shells.) If a special built-in utility encountering a syntax error does not  
2646       abort the shell, its exit value shall be non-zero.
- 2647     2. Variable assignments specified with special built-in utilities remain in effect after the  
2648       built-in completes; this shall not be the case with a regular built-in or other utility.

2649 The special built-in utilities in this section need not be provided in a manner accessible via the  
2650 `exec` family of functions defined in the System Interfaces volume of IEEE Std 1003.1-2001.

2651 Some of the special built-ins are described as conforming to the Base Definitions volume of  
2652 IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines. For those that are not, the  
2653 requirement in Section 1.11 (on page 20) that "--" be recognized as a first argument to be  
2654 discarded does not apply and a conforming application shall not use that argument.

**2655 NAME**

2656       **break** — exit from for, while, or until loop

**2657 SYNOPSIS**

2658       **break** [*n*]

**2659 DESCRIPTION**

2660       The *break* utility shall exit from the smallest enclosing **for**, **while**, or **until** loop, if any; or from the  
2661       *n*th enclosing loop if *n* is specified. The value of *n* is an unsigned decimal integer greater than or  
2662       equal to 1. The default shall be equivalent to *n*=1. If *n* is greater than the number of enclosing  
2663       loops, the outermost enclosing loop shall be exited. Execution shall continue with the command  
2664       immediately following the loop.

**2665 OPTIONS**

2666       None.

**2667 OPERANDS**

2668       None.

**2669 STDIN**

2670       None.

**2671 INPUT FILES**

2672       None.

**2673 ENVIRONMENT VARIABLES**

2674       None.

**2675 ASYNCHRONOUS EVENTS**

2676       None.

**2677 STDOUT**

2678       None.

**2679 STDERR**

2680       None.

**2681 OUTPUT FILES**

2682       None.

**2683 EXTENDED DESCRIPTION**

2684       None.

**2685 EXIT STATUS**

2686       0      Successful completion.

2687       >0     The *n* value was not an unsigned decimal integer greater than or equal to 1.

**2688 CONSEQUENCES OF ERRORS**

2689       None.

2690 **APPLICATION USAGE**

2691 None.

2692 **EXAMPLES**2693     for i in \* do  
2694         if test -d "\$i" then break fi done2695 **RATIONALE**2696 In early proposals, consideration was given to expanding the syntax of *break* and *continue* to refer  
2697 to a label associated with the appropriate loop as a preferable alternative to the *n* method.  
2698 However, this volume of IEEE Std 1003.1-2001 does reserve the name space of command names  
2699 ending with a colon. It is anticipated that a future implementation could take advantage of this  
2700 and provide something like:2701     outoffloop: for i in a b c d e  
2702         do  
2703             for j in 0 1 2 3 4 5 6 7 8 9  
2704                 do  
2705                     if test -r "\${i}\${j}"  
2706                     then break outoffloop  
2707                     fi  
2708                 done  
2709         done

2710 and that this might be standardized after implementation experience is achieved.

2711 **FUTURE DIRECTIONS**

2712 None.

2713 **SEE ALSO**

2714 Section 2.14 (on page 64)

2715 **CHANGE HISTORY**

2716 None.

**2717 NAME**

2718 colon — null utility

**2719 SYNOPSIS**

2720 : [argument ...]

**2721 DESCRIPTION**

2722 This utility shall only expand command *arguments*. It is used when a command is needed, as in  
2723 the **then** condition of an **if** command, but nothing is to be done by the command.

**2724 OPTIONS**

2725 None.

**2726 OPERANDS**

2727 None.

**2728 STDIN**

2729 None.

**2730 INPUT FILES**

2731 None.

**2732 ENVIRONMENT VARIABLES**

2733 None.

**2734 ASYNCHRONOUS EVENTS**

2735 None.

**2736 STDOUT**

2737 None.

**2738 STDERR**

2739 None.

**2740 OUTPUT FILES**

2741 None.

**2742 EXTENDED DESCRIPTION**

2743 None.

**2744 EXIT STATUS**

2745 Zero.

**2746 CONSEQUENCES OF ERRORS**

2747 None.

**2748 APPLICATION USAGE**

2749 None.

**2750 EXAMPLES**

2751 : \${X=abc}  
2752 if false  
2753 then :  
2754 else echo \$X  
2755 fi  
2756 abc

2757 As with any of the special built-ins, the null utility can also have variable assignments and  
2758 redirections associated with it, such as:

2759           `x=y : > z`  
2760           which sets variable **x** to the value **y** (so that it persists after the null utility completes) and creates  
2761           or truncates file **z**.

2762       **RATIONALE**

2763           None.

2764       **FUTURE DIRECTIONS**

2765           None.

2766       **SEE ALSO**

2767           Section 2.14 (on page 64)

2768       **CHANGE HISTORY**

2769           None.

**2770 NAME**

2771 continue — continue for, while, or until loop

**2772 SYNOPSIS**

2773 continue [n]

**2774 DESCRIPTION**

2775 The *continue* utility shall return to the top of the smallest enclosing **for**, **while**, or **until** loop, or to  
2776 the top of the *n*th enclosing loop, if *n* is specified. This involves repeating the condition list of a  
2777 **while** or **until** loop or performing the next assignment of a **for** loop, and re-executing the loop if  
2778 appropriate.

2779 The value of *n* is a decimal integer greater than or equal to 1. The default shall be equivalent to  
2780 *n*=1. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be  
2781 used.

**2782 OPTIONS**

2783 None.

**2784 OPERANDS**

2785 None.

**2786 STDIN**

2787 None.

**2788 INPUT FILES**

2789 None.

**2790 ENVIRONMENT VARIABLES**

2791 None.

**2792 ASYNCHRONOUS EVENTS**

2793 None.

**2794 STDOUT**

2795 None.

**2796 STDERR**

2797 None.

**2798 OUTPUT FILES**

2799 None.

**2800 EXTENDED DESCRIPTION**

2801 None.

**2802 EXIT STATUS**

2803 0 Successful completion.

2804 >0 The *n* value was not an unsigned decimal integer greater than or equal to 1.

**2805 CONSEQUENCES OF ERRORS**

2806 None.

2807 **APPLICATION USAGE**

2808       None.

2809 **EXAMPLES**

```
2810 for i in *
2811 do
2812 if test -d "$i"
2813 then continue
2814 fi
2815 echo "\"$i\" is not a directory.
2816 done
```

2817 **RATIONALE**

2818       None.

2819 **FUTURE DIRECTIONS**

2820       None.

2821 **SEE ALSO**

2822       Section 2.14 (on page 64)

2823 **CHANGE HISTORY**

2824       None.

**2825 NAME**

2826       dot — execute commands in the current environment

**2827 SYNOPSIS**

2828       . *file*

**2829 DESCRIPTION**

2830       The shell shall execute commands from the *file* in the current environment.

2831       If *file* does not contain a slash, the shell shall use the search path specified by *PATH* to find the  
2832       directory containing *file*. Unlike normal command search, however, the file searched for by the  
2833       *dot* utility need not be executable. If no readable file is found, a non-interactive shell shall abort;  
2834       an interactive shell shall write a diagnostic message to standard error, but this condition shall  
2835       not be considered a syntax error.

**2836 OPTIONS**

2837       None.

**2838 OPERANDS**

2839       None.

**2840 STDIN**

2841       None.

**2842 INPUT FILES**

2843       None.

**2844 ENVIRONMENT VARIABLES**

2845       None.

**2846 ASYNCHRONOUS EVENTS**

2847       None.

**2848 STDOUT**

2849       None.

**2850 STDERR**

2851       The standard error shall be used only for diagnostic messages.

**2852 OUTPUT FILES**

2853       None.

**2854 EXTENDED DESCRIPTION**

2855       None.

**2856 EXIT STATUS**

2857       Returns the value of the last command executed, or a zero exit status if no command is executed.

**2858 CONSEQUENCES OF ERRORS**

2859       None.

2860 **APPLICATION USAGE**

2861 None.

2862 **EXAMPLES**

```
2863 cat foobar
2864 foo=hello bar=world
2865 . foobar
2866 echo $foo $bar
2867 hello world
```

2868 **RATIONALE**

2869 Some older implementations searched the current directory for the *file*, even if the value of *PATH*  
2870 disallowed it. This behavior was omitted from this volume of IEEE Std 1003.1-2001 due to  
2871 concerns about introducing the susceptibility to trojan horses that the user might be trying to  
2872 avoid by leaving **dot** out of *PATH*.

2873 The KornShell version of *dot* takes optional arguments that are set to the positional parameters.  
2874 This is a valid extension that allows a *dot* script to behave identically to a function.

2875 **FUTURE DIRECTIONS**

2876 None.

2877 **SEE ALSO**

2878 Section 2.14 (on page 64)

2879 **CHANGE HISTORY**

2880 None.

**2881 NAME**

2882 eval — construct command by concatenating arguments

**2883 SYNOPSIS**

2884 eval [argument ...]

**2885 DESCRIPTION**

2886 The *eval* utility shall construct a command by concatenating *arguments* together, separating each  
2887 with a <space>. The constructed command shall be read and executed by the shell.

**2888 OPTIONS**

2889 None.

**2890 OPERANDS**

2891 None.

**2892 STDIN**

2893 None.

**2894 INPUT FILES**

2895 None.

**2896 ENVIRONMENT VARIABLES**

2897 None.

**2898 ASYNCHRONOUS EVENTS**

2899 None.

**2900 STDOUT**

2901 None.

**2902 STDERR**

2903 None.

**2904 OUTPUT FILES**

2905 None.

**2906 EXTENDED DESCRIPTION**

2907 None.

**2908 EXIT STATUS**

2909 If there are no *arguments*, or only null arguments, *eval* shall return a zero exit status; otherwise, it  
2910 shall return the exit status of the command defined by the string of concatenated *arguments*  
2911 separated by <space>s.

**2912 CONSEQUENCES OF ERRORS**

2913 None.

**2914 APPLICATION USAGE**

2915 None.

**2916 EXAMPLES**

```
2917 foo=10 x=foo
2918 y='$'$x
2919 echo $y
2920 $foo
2921 eval y='$'$x
2922 echo $y
2923 10
```

2924 **RATIONALE**

2925       None.

2926 **FUTURE DIRECTIONS**

2927       None.

2928 **SEE ALSO**

2929       Section 2.14 (on page 64)

2930 **CHANGE HISTORY**

2931       None.

**2932 NAME**

2933 exec — execute commands and open, close, or copy file descriptors

**2934 SYNOPSIS**

2935 exec [*command* [*argument* . . .]]

**2936 DESCRIPTION**

2937 The *exec* utility shall open, close, and/or copy file descriptors as specified by any redirections as  
2938 part of the command.

2939 If *exec* is specified without *command* or *arguments*, and any file descriptors with numbers greater  
2940 than 2 are opened with associated redirection statements, it is unspecified whether those file  
2941 descriptors remain open when the shell invokes another utility. Scripts concerned that child  
2942 shells could misuse open file descriptors can always close them explicitly, as shown in one of the  
2943 following examples.

2944 If *exec* is specified with *command*, it shall replace the shell with *command* without creating a new  
2945 process. If *arguments* are specified, they shall be arguments to *command*. Redirection affects the  
2946 current shell execution environment.

**2947 OPTIONS**

2948 None.

**2949 OPERANDS**

2950 None.

**2951 STDIN**

2952 None.

**2953 INPUT FILES**

2954 None.

**2955 ENVIRONMENT VARIABLES**

2956 None.

**2957 ASYNCHRONOUS EVENTS**

2958 None.

**2959 STDOUT**

2960 None.

**2961 STDERR**

2962 None.

**2963 OUTPUT FILES**

2964 None.

**2965 EXTENDED DESCRIPTION**

2966 None.

**2967 EXIT STATUS**

2968 If *command* is specified, *exec* shall not return to the shell; rather, the exit status of the process shall  
2969 be the exit status of the program implementing *command*, which overlaid the shell. If *command* is  
2970 not found, the exit status shall be 127. If *command* is found, but it is not an executable utility, the  
2971 exit status shall be 126. If a redirection error occurs (see Section 2.8.1 (on page 46)), the shell shall  
2972 exit with a value in the range 1–125. Otherwise, *exec* shall return a zero exit status.

2973 **CONSEQUENCES OF ERRORS**

2974 None.

2975 **APPLICATION USAGE**

2976 None.

2977 **EXAMPLES**2978 Open *readfile* as file descriptor 3 for reading:

2979 exec 3&lt; readfile

2980 Open *writefile* as file descriptor 4 for writing:

2981 exec 4&gt; writefile

2982 Make file descriptor 5 a copy of file descriptor 0:

2983 exec 5&lt;&amp;0

2984 Close file descriptor 3:

2985 exec 3&lt;&amp;-

2986 Cat the file **maggie** by replacing the current shell with the *cat* utility:

2987 exec cat maggie

2988 **RATIONALE**

2989 Most historical implementations were not conformant in that:

2990 foo=bar exec cmd

2991 did not pass **foo** to **cmd**.2992 **FUTURE DIRECTIONS**

2993 None.

2994 **SEE ALSO**

2995 Section 2.14 (on page 64)

2996 **CHANGE HISTORY**

2997 None.

2998 **NAME**

2999       exit — cause the shell to exit

3000 **SYNOPSIS**

3001       exit [n]

3002 **DESCRIPTION**3003       The *exit* utility shall cause the shell to exit with the exit status specified by the unsigned decimal  
3004       integer *n*. If *n* is specified, but its value is not between 0 and 255 inclusively, the exit status is  
3005       undefined.3006       A *trap* on **EXIT** shall be executed before the shell terminates, except when the *exit* utility is  
3007       invoked in that *trap* itself, in which case the shell shall exit immediately.3008 **OPTIONS**

3009       None.

3010 **OPERANDS**

3011       None.

3012 **STDIN**

3013       None.

3014 **INPUT FILES**

3015       None.

3016 **ENVIRONMENT VARIABLES**

3017       None.

3018 **ASYNCHRONOUS EVENTS**

3019       None.

3020 **STDOUT**

3021       None.

3022 **STDERR**

3023       None.

3024 **OUTPUT FILES**

3025       None.

3026 **EXTENDED DESCRIPTION**

3027       None.

3028 **EXIT STATUS**3029       The exit status shall be *n*, if specified. Otherwise, the value shall be the exit value of the last  
3030       command executed, or zero if no command was executed. When *exit* is executed in a *trap* action,  
3031       the last command is considered to be the command that executed immediately preceding the  
3032       *trap* action.3033 **CONSEQUENCES OF ERRORS**

3034       None.

**3035 APPLICATION USAGE**

3036 None.

**3037 EXAMPLES**

3038 Exit with a *true* value:

3039 exit 0

3040 Exit with a *false* value:

3041 exit 1

**3042 RATIONALE**

3043 As explained in other sections, certain exit status values have been reserved for special uses and  
3044 should be used by applications only for those purposes:

3045 126 A file to be executed was found, but it was not an executable utility.

3046 127 A utility to be executed was not found.

3047 >128 A command was interrupted by a signal.

**3048 FUTURE DIRECTIONS**

3049 None.

**3050 SEE ALSO**

3051 Section 2.14 (on page 64)

**3052 CHANGE HISTORY**

3053 None.

3054 **NAME**

3055        export — set the export attribute for variables

3056 **SYNOPSIS**

3057        export name[=word] . . .

3058        export -p

3059 **DESCRIPTION**

3060        The shell shall give the *export* attribute to the variables corresponding to the specified *names*,  
3061        which shall cause them to be in the environment of subsequently executed commands.

3062        The *export* special built-in shall support the Base Definitions volume of IEEE Std 1003.1-2001,  
3063        Section 12.2, Utility Syntax Guidelines.

3064        When **-p** is specified, *export* shall write to the standard output the names and values of all  
3065        exported variables, in the following format:

3066        "export %s=%s\n", <name>, <value>

3067        if *name* is set, and:

3068        "export %s\n", <name>

3069        if *name* is unset.

3070        The shell shall format the output, including the proper use of quoting, so that it is suitable for  
3071        reinput to the shell as commands that achieve the same exporting results, except:

- 3072        1. Read-only variables with values cannot be reset.
- 3073        2. Variables that were unset at the time they were output need not be reset to the unset state  
3074            if a value is assigned to the variable between the time the state was saved and the time at  
3075            which the saved output is reinput to the shell.

3076        When no arguments are given, the results are unspecified.

3077 **OPTIONS**

3078        None.

3079 **OPERANDS**

3080        None.

3081 **STDIN**

3082        None.

3083 **INPUT FILES**

3084        None.

3085 **ENVIRONMENT VARIABLES**

3086        None.

3087 **ASYNCHRONOUS EVENTS**

3088        None.

3089 **STDOUT**

3090        None.

3091 **STDERR**

3092        None.

**3093 OUTPUT FILES**

3094 None.

**3095 EXTENDED DESCRIPTION**

3096 None.

**3097 EXIT STATUS**

3098 Zero.

**3099 CONSEQUENCES OF ERRORS**

3100 None.

**3101 APPLICATION USAGE**

3102 None.

**3103 EXAMPLES**

3104 Export *PWD* and *HOME* variables:

```
3105 export PWD HOME
```

3106 Set and export the *PATH* variable:

```
3107 export PATH=/local/bin:$PATH
```

3108 Save and restore all exported variables:

```
3109 export -p > temp-file
3110 unset a lot of variables
3111 ... processing
3112 . temp-file
```

**3113 RATIONALE**

3114 Some historical shells use the no-argument case as the functional equivalent of what is required  
3115 here with **-p**. This feature was left unspecified because it is not historical practice in all shells,  
3116 and some scripts may rely on the now-unspecified results on their implementations. Attempts to  
3117 specify the **-p** output as the default case were unsuccessful in achieving consensus. The **-p**  
3118 option was added to allow portable access to the values that can be saved and then later restored  
3119 using; for example, a *dot* script.

**3120 FUTURE DIRECTIONS**

3121 None.

**3122 SEE ALSO**

3123 Section 2.14 (on page 64)

**3124 CHANGE HISTORY****3125 Issue 6**

3126 IEEE PASC Interpretation 1003.2 #203 is applied, clarifying the format when a variable is unset.

3127 **NAME**

3128        readonly — set the readonly attribute for variables

3129 **SYNOPSIS**

3130        `readonly name[=word]...`

3131        `readonly -p`

3132 **DESCRIPTION**

3133        The variables whose *names* are specified shall be given the *readonly* attribute. The values of  
3134        variables with the *readonly* attribute cannot be changed by subsequent assignment, nor can those  
3135        variables be unset by the *unset* utility.

3136        The *readonly* special built-in shall support the Base Definitions volume of IEEE Std 1003.1-2001,  
3137        Section 12.2, Utility Syntax Guidelines.

3138        When **-p** is specified, *readonly* writes to the standard output the names and values of all read-  
3139        only variables, in the following format:

3140        `"readonly %s=%s\n", <name>, <value>`

3141        if *name* is set, and

3142        `"readonly %s\n", <name>`

3143        if *name* is unset.

3144        The shell shall format the output, including the proper use of quoting, so that it is suitable for  
3145        reinput to the shell as commands that achieve the same value and *readonly* attribute-setting  
3146        results in a shell execution environment in which:

- 3147        1. Variables with values at the time they were output do not have the *readonly* attribute set.
- 3148        2. Variables that were unset at the time they were output do not have a value at the time at  
3149              which the saved output is reinput to the shell.

3150        When no arguments are given, the results are unspecified.

3151 **OPTIONS**

3152        None.

3153 **OPERANDS**

3154        None.

3155 **STDIN**

3156        None.

3157 **INPUT FILES**

3158        None.

3159 **ENVIRONMENT VARIABLES**

3160        None.

3161 **ASYNCHRONOUS EVENTS**

3162        None.

3163 **STDOUT**

3164        None.

**3165 STDRERR**

3166 None.

**3167 OUTPUT FILES**

3168 None.

**3169 EXTENDED DESCRIPTION**

3170 None.

**3171 EXIT STATUS**

3172 Zero.

**3173 CONSEQUENCES OF ERRORS**

3174 None.

**3175 APPLICATION USAGE**

3176 None.

**3177 EXAMPLES**

3178 `readonly HOME PWD`

**3179 RATIONALE**

3180 Some historical shells preserve the *readonly* attribute across separate invocations. This volume of  
3181 IEEE Std 1003.1-2001 allows this behavior, but does not require it.

3182 The **-p** option allows portable access to the values that can be saved and then later restored  
3183 using, for example, a *dot* script. Also see the RATIONALE for *export* (on page 79) for a  
3184 description of the no-argument and **-p** output cases and a related example.

3185 Read-only functions were considered, but they were omitted as not being historical practice or  
3186 particularly useful. Furthermore, functions must not be read-only across invocations to preclude  
3187 “spoofing” (spoofing is the term for the practice of creating a program that acts like a well-  
3188 known utility with the intent of subverting the real intent of the user) of administrative or  
3189 security-relevant (or security-conscious) shell scripts.

**3190 FUTURE DIRECTIONS**

3191 None.

**3192 SEE ALSO**

3193 Section 2.14 (on page 64)

**3194 CHANGE HISTORY****3195 Issue 6**

3196 IEEE PASC Interpretation 1003.2 #203 is applied, clarifying the format when a variable is unset.

**3197 NAME**

3198        return — return from a function

**3199 SYNOPSIS**

3200        return [n]

**3201 DESCRIPTION**

3202        The *return* utility shall cause the shell to stop executing the current function or *dot* script. If the  
3203        shell is not currently executing a function or *dot* script, the results are unspecified.

**3204 OPTIONS**

3205        None.

**3206 OPERANDS**

3207        None.

**3208 STDIN**

3209        None.

**3210 INPUT FILES**

3211        None.

**3212 ENVIRONMENT VARIABLES**

3213        None.

**3214 ASYNCHRONOUS EVENTS**

3215        None.

**3216 STDOUT**

3217        None.

**3218 STDERR**

3219        None.

**3220 OUTPUT FILES**

3221        None.

**3222 EXTENDED DESCRIPTION**

3223        None.

**3224 EXIT STATUS**

3225        The value of the special parameter '?' shall be set to *n*, an unsigned decimal integer, or to the  
3226        exit status of the last command executed if *n* is not specified. If the value of *n* is greater than 255,  
3227        the results are undefined. When *return* is executed in a *trap* action, the last command is  
3228        considered to be the command that executed immediately preceding the *trap* action.

**3229 CONSEQUENCES OF ERRORS**

3230        None.

**3231 APPLICATION USAGE**

3232        None.

**3233 EXAMPLES**

3234        None.

**3235 RATIONALE**

3236        The behavior of *return* when not in a function or *dot* script differs between the System V shell  
3237        and the KornShell. In the System V shell this is an error, whereas in the KornShell, the effect is  
3238        the same as *exit*.

3239        The results of returning a number greater than 255 are undefined because of differing practices  
3240        in the various historical implementations. Some shells AND out all but the low-order 8 bits;  
3241        others allow larger values, but not of unlimited size.

3242        See the discussion of appropriate exit status values under *exit* (on page 77).

3243 **FUTURE DIRECTIONS**

3244        None.

3245 **SEE ALSO**

3246        Section 2.14 (on page 64)

3247 **CHANGE HISTORY**

3248        None.

3249 **NAME**

3250       set — set or unset options and positional parameters

3251 **SYNOPSIS**

3252 XSI       set [-abCefmnuvx][-h][-o option][argument...]

3253 XSI       set [+abCefmnuvx][+h][+o option][argument...]

3254       set -- [argument...]

3255       set -o

3256       set +o

3257 **DESCRIPTION**

3258 If no *options* or *arguments* are specified, *set* shall write the names and values of all shell variables  
 3259 in the collation sequence of the current locale. Each *name* shall start on a separate line, using the  
 3260 format:

3261       "%s=%s\n", <name>, <value>

3262 The *value* string shall be written with appropriate quoting; see the description of shell quoting in  
 3263 Section 2.2 (on page 30). The output shall be suitable for reinput to the shell, setting or resetting,  
 3264 as far as possible, the variables that are currently set; read-only variables cannot be reset.

3265 When options are specified, they shall set or unset attributes of the shell, as described below.  
 3266 When *arguments* are specified, they cause positional parameters to be set or unset, as described  
 3267 below. Setting or unsetting attributes and positional parameters are not necessarily related  
 3268 actions, but they can be combined in a single invocation of *set*.

3269 The *set* special built-in shall support the Base Definitions volume of IEEE Std 1003.1-2001,  
 3270 Section 12.2, Utility Syntax Guidelines except that options can be specified with either a leading  
 3271 hyphen (meaning enable the option) or plus sign (meaning disable it) unless otherwise specified.

3272 Implementations shall support the options in the following list in both their hyphen and plus-  
 3273 sign forms. These options can also be specified as options to *sh*.

3274 **-a** When this option is on, the *export* attribute shall be set for each variable to which an  
 3275 assignment is performed; see the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 3276 4.21, Variable Assignment. If the assignment precedes a utility name in a command, the  
 3277 *export* attribute shall not persist in the current execution environment after the utility  
 3278 completes, with the exception that preceding one of the special built-in utilities causes the  
 3279 *export* attribute to persist after the built-in has completed. If the assignment does not  
 3280 precede a utility name in the command, or if the assignment is a result of the operation of  
 3281 the *getopts* or *read* utilities, the *export* attribute shall persist until the variable is unset.

3282 **-b** This option shall be supported if the implementation supports the User Portability Utilities  
 3283 option. It shall cause the shell to notify the user asynchronously of background job  
 3284 completions. The following message is written to standard error:

3285       "[%d] %c %s%s\n", <job-number>, <current>, <status>, <job-name>

3286 where the fields shall be as follows:

3287       <current>       The character '+' identifies the job that would be used as a default for  
 3288 the *fg* or *bg* utilities; this job can also be specified using the *job\_id* "%+" or  
 3289 "%%". The character '-' identifies the job that would become the default  
 3290 if the current default job were to exit; this job can also be specified using  
 3291 the *job\_id* "%-". For other jobs, this field is a <space>. At most one job  
 3292 can be identified with '+' and at most one job can be identified with '-'.

|          |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3293     |                  | If there is any suspended job, then the current job shall be a suspended job. If there are at least two suspended jobs, then the previous job also shall be a suspended job.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3296     | <job-number>     | A number that can be used to identify the process group to the <i>wait</i> , <i>fg</i> , <i>bg</i> , and <i>kill</i> utilities. Using these utilities, the job can be identified by prefixing the job number with '%'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3299     | <status>         | Unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3300     | <job-name>       | Unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3301     |                  | When the shell notifies the user a job has been completed, it may remove the job's process ID from the list of those known in the current shell execution environment; see Section 2.9.3.1 (on page 50). Asynchronous notification shall not be enabled by default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3304     | <b>-C</b>        | (Uppercase C.) Prevent existing files from being overwritten by the shell's '>' redirection operator (see Section 2.7.2 (on page 44)); the "> " redirection operator shall override this <i>noclobber</i> option for an individual file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3307     | <b>-e</b>        | When this option is on, if a simple command fails for any of the reasons listed in Section 2.8.1 (on page 46) or returns an exit status value >0, and is not part of the compound list following a <b>while</b> , <b>until</b> , or <b>if</b> keyword, and is not a part of an AND or OR list, and is not a pipeline preceded by the ! reserved word, then the shell shall immediately exit.                                                                                                                                                                                                                                                                                                                                                                                         |
| 3311     | <b>-f</b>        | The shell shall disable pathname expansion.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 3312 XSI | <b>-h</b>        | Locate and remember utilities invoked by functions as those functions are defined (the utilities are normally located when the function is executed).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 3314     | <b>-m</b>        | This option shall be supported if the implementation supports the User Portability Utilities option. All jobs shall be run in their own process groups. Immediately before the shell issues a prompt after completion of the background job, a message reporting the exit status of the background job shall be written to standard error. If a foreground job stops, the shell shall write a message to standard error to that effect, formatted as described by the <i>jobs</i> utility. In addition, if a job changes status other than exiting (for example, if it stops for input or output or is stopped by a SIGSTOP signal), the shell shall write a similar message immediately prior to writing the next prompt. This option is enabled by default for interactive shells. |
| 3323     | <b>-n</b>        | The shell shall read commands but does not execute them; this can be used to check for shell script syntax errors. An interactive shell may ignore this option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 3325     | <b>-o</b>        | Write the current settings of the options to standard output in an unspecified format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3326     | <b>+o</b>        | Write the current option settings to standard output in a format that is suitable for reinput to the shell as commands that achieve the same options settings.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 3328     | <b>-o option</b> | This option is supported if the system supports the User Portability Utilities option. It shall set various options, many of which shall be equivalent to the single option letters. The following values of <i>option</i> shall be supported:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 3332     | <b>allexport</b> | Equivalent to <b>-a</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3333     | <b>errexit</b>   | Equivalent to <b>-e</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3334     | <b>ignoreeof</b> | Prevent an interactive shell from exiting on end-of-file. This setting prevents accidental logouts when <control>-D is entered. A user shall explicitly <i>exit</i> to leave the interactive shell.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3335     |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 3336     |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

|      |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3337 | <i>monitor</i>   | Equivalent to <b>-m</b> . This option is supported if the system supports the User Portability Utilities option.                                                                                                                                                                                                                                                                                          |
| 3338 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3339 | <i>noclobber</i> | Equivalent to <b>-C</b> (uppercase C).                                                                                                                                                                                                                                                                                                                                                                    |
| 3340 | <i>noglob</i>    | Equivalent to <b>-f</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3341 | <i>noexec</i>    | Equivalent to <b>-n</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3342 | <i>nolog</i>     | Prevent the entry of function definitions into the command history; see <b>Command History List</b> (on page 851).                                                                                                                                                                                                                                                                                        |
| 3343 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3344 | <i>notify</i>    | Equivalent to <b>-b</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3345 | <i>nounset</i>   | Equivalent to <b>-u</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3346 | <i>verbose</i>   | Equivalent to <b>-v</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3347 | <i>vi</i>        | Allow shell command line editing using the built-in <i>vi</i> editor. Enabling <i>vi</i> mode shall disable any other command line editing mode provided as an implementation extension.                                                                                                                                                                                                                  |
| 3348 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3349 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3350 |                  | It need not be possible to set <i>vi</i> mode on for certain block-mode terminals.                                                                                                                                                                                                                                                                                                                        |
| 3351 | <i>xtrace</i>    | Equivalent to <b>-x</b> .                                                                                                                                                                                                                                                                                                                                                                                 |
| 3352 | <b>-u</b>        | The shell shall write a message to standard error when it tries to expand a variable that is not set and immediately exit. An interactive shell shall not exit.                                                                                                                                                                                                                                           |
| 3353 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3354 | <b>-v</b>        | The shell shall write its input to standard error as it is read.                                                                                                                                                                                                                                                                                                                                          |
| 3355 | <b>-x</b>        | The shell shall write to standard error a trace for each command after it expands the command and before it executes it. It is unspecified whether the command that turns tracing off is traced.                                                                                                                                                                                                          |
| 3356 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3357 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3358 |                  | The default for all these options shall be off (unset) unless stated otherwise in the description of the option or unless the shell was invoked with them on; see <i>sh</i> .                                                                                                                                                                                                                             |
| 3359 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3360 |                  | The remaining arguments shall be assigned in order to the positional parameters. The special parameter ' <b>#</b> ' shall be set to reflect the number of positional parameters. All positional parameters shall be unset before any new values are assigned.                                                                                                                                             |
| 3361 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3362 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3363 |                  | The special argument " <b>--</b> " immediately following the <i>set</i> command name can be used to delimit the arguments if the first argument begins with '+' or '-', or to prevent inadvertent listing of all shell variables when there are no arguments. The command <i>set --</i> without <i>argument</i> shall unset all positional parameters and set the special parameter ' <b>#</b> ' to zero. |
| 3364 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3365 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3366 |                  |                                                                                                                                                                                                                                                                                                                                                                                                           |

## 3367 OPTIONS

3368 None.

## 3369 OPERANDS

3370 None.

## 3371 STDIN

3372 None.

## 3373 INPUT FILES

3374 None.

3375 **ENVIRONMENT VARIABLES**

3376 None.

3377 **ASYNCHRONOUS EVENTS**

3378 None.

3379 **STDOUT**

3380 None.

3381 **STDERR**

3382 None.

3383 **OUTPUT FILES**

3384 None.

3385 **EXTENDED DESCRIPTION**

3386 None.

3387 **EXIT STATUS**

3388 Zero.

3389 **CONSEQUENCES OF ERRORS**

3390 None.

3391 **APPLICATION USAGE**

3392 None.

3393 **EXAMPLES**

3394 Write out all variables and their values:

3395 set

3396 Set \$1, \$2, and \$3 and set "\$#" to 3:

3397 set c a b

3398 Turn on the -x and -v options:

3399 set -xv

3400 Unset all positional parameters:

3401 set --

3402 Set \$1 to the value of x, even if it begins with '-' or '+':

3403 set -- "\$x"

3404 Set the positional parameters to the expansion of x, even if x expands with a leading '-' or '+':

3405 set -- \$x

3406 **RATIONALE**

3407 The `set --` form is listed specifically in the SYNOPSIS even though this usage is implied by the  
3408 Utility Syntax Guidelines. The explanation of this feature removes any ambiguity about whether  
3409 the `set --` form might be misinterpreted as being equivalent to `set` without any options or  
3410 arguments. The functionality of this form has been adopted from the KornShell. In System V, `set`  
3411 -- only unsets parameters if there is at least one argument; the only way to unset all parameters  
3412 is to use `shift`. Using the KornShell version should not affect System V scripts because there  
3413 should be no reason to issue it without arguments deliberately; if it were issued as, for example:

3414 set -- "\$@"

3415 and there were in fact no arguments resulting from "\$@", unsetting the parameters would have  
3416 no result.

3417 The *set +* form in early proposals was omitted as being an unnecessary duplication of *set* alone  
3418 and not widespread historical practice.

3419 The *noclobber* option was changed to allow *set -C* as well as the *set -o noclobber* option. The  
3420 single-letter version was added so that the historical "\$-" paradigm would not be broken; see  
3421 Section 2.5.2 (on page 34).

3422 The **-h** flag is related to command name hashing and is only required on XSI-conformant  
3423 systems.

3424 The following *set* flags were omitted intentionally with the following rationale:

3425 **-k** The **-k** flag was originally added by the author of the Bourne shell to make it easier for  
3426 users of pre-release versions of the shell. In early versions of the Bourne shell the construct  
3427 *set name=value* had to be used to assign values to shell variables. The problem with **-k** is  
3428 that the behavior affects parsing, virtually precluding writing any compilers. To explain the  
3429 behavior of **-k**, it is necessary to describe the parsing algorithm, which is implementation-  
3430 defined. For example:

3431    *set -k; echo name=value*

3432 and:

3433    *set -k*  
3434    *echo name=value*

3435 behave differently. The interaction with functions is even more complex. What is more, the  
3436 **-k** flag is never needed, since the command line could have been reordered.

3437 **-t** The **-t** flag is hard to specify and almost never used. The only known use could be done  
3438 with here-documents. Moreover, the behavior with *ksh* and *sh* differs. The reference page  
3439 says that it exits after reading and executing one command. What is one command? If the  
3440 input is *date;date*, *sh* executes both *date* commands while *ksh* does only the first.

3441 Consideration was given to rewriting *set* to simplify its confusing syntax. A specific suggestion  
3442 was that the *unset* utility should be used to unset options instead of using the non-*getopt()*-able  
3443 *+option* syntax. However, the conclusion was reached that the historical practice of using *+option*  
3444 was satisfactory and that there was no compelling reason to modify such widespread historical  
3445 practice.

3446 The **-o** option was adopted from the KornShell to address user needs. In addition to its generally  
3447 friendly interface, **-o** is needed to provide the *vi* command line editing mode, for which  
3448 historical practice yields no single-letter option name. (Although it might have been possible to  
3449 invent such a letter, it was recognized that other editing modes would be developed and **-o**  
3450 provides ample name space for describing such extensions.)

3451 Historical implementations are inconsistent in the format used for **-o** option status reporting.  
3452 The **+o** format without an option-argument was added to allow portable access to the options  
3453 that can be saved and then later restored using, for instance, a dot script.

3454 Historically, *sh* did trace the command *set +x*, but *ksh* did not.

3455 The *ignoreeof* setting prevents accidental logouts when the end-of-file character (typically  
3456 <control>-D) is entered. A user shall explicitly *exit* to leave the interactive shell.

3457 The *set -m* option was added to apply only to the UPE because it applies primarily to interactive  
3458 use, not shell script applications.

3459       The ability to do asynchronous notification became available in the 1988 version of the  
3460       KornShell. To have it occur, the user had to issue the command:

3461       trap "jobs -n" CLD

3462       The C shell provides two different levels of an asynchronous notification capability. The  
3463       environment variable *notify* is analogous to what is done in *set -b* or *set -o notify*. When set, it  
3464       notifies the user immediately of background job completions. When unset, this capability is  
3465       turned off.

3466       The other notification ability comes through the built-in utility *notify*. The syntax is:

3467       *notify* [%job ... ]

3468       By issuing *notify* with no operands, it causes the C shell to notify the user asynchronously when  
3469       the state of the current job changes. If given operands, *notify* asynchronously informs the user of  
3470       changes in the states of the specified jobs.

3471       To add asynchronous notification to the POSIX shell, neither the KornShell extensions to *trap*,  
3472       nor the C shell *notify* environment variable seemed appropriate (*notify* is not a proper POSIX  
3473       environment variable name).

3474       The *set -b* option was selected as a compromise.

3475       The *notify* built-in was considered to have more functionality than was required for simple  
3476       asynchronous notification.

## 3477 FUTURE DIRECTIONS

3478       None.

## 3479 SEE ALSO

3480       Section 2.14 (on page 64)

## 3481 CHANGE HISTORY

### 3482 Issue 6

3483       The obsolescent *set* command name followed by '-' has been removed.

3484       The following new requirements on POSIX implementations derive from alignment with the  
3485       Single UNIX Specification:

- The *nolog* option is added to *set -o*.

3487       IEEE PASC Interpretation 1003.2 #167 is applied, clarifying that the options default also takes  
3488       into account the description of the option.

**3489 NAME**

3490 shift — shift positional parameters

**3491 SYNOPSIS**

3492 shift [n]

**3493 DESCRIPTION**

3494 The positional parameters shall be shifted. Positional parameter 1 shall be assigned the value of parameter  $(1+n)$ , parameter 2 shall be assigned the value of parameter  $(2+n)$ , and so on. The parameters represented by the numbers "\$#" down to "\$#-n+1" shall be unset, and the parameter '#' is updated to reflect the new number of positional parameters.

3498 The value n shall be an unsigned decimal integer less than or equal to the value of the special parameter '#'. If n is not given, it shall be assumed to be 1. If n is 0, the positional and special parameters are not changed.

**3501 OPTIONS**

3502 None.

**3503 OPERANDS**

3504 None.

**3505 STDIN**

3506 None.

**3507 INPUT FILES**

3508 None.

**3509 ENVIRONMENT VARIABLES**

3510 None.

**3511 ASYNCHRONOUS EVENTS**

3512 None.

**3513 STDOUT**

3514 None.

**3515 STDERR**

3516 None.

**3517 OUTPUT FILES**

3518 None.

**3519 EXTENDED DESCRIPTION**

3520 None.

**3521 EXIT STATUS**

3522 The exit status is >0 if  $n > \$\#$ ; otherwise, it is zero.

**3523 CONSEQUENCES OF ERRORS**

3524 None.

**3525 APPLICATION USAGE**

3526       None.

**3527 EXAMPLES**

```
3528 $ set a b c d e
3529 $ shift 2
3530 $ echo $*
3531 c d e
```

**3532 RATIONALE**

3533       None.

**3534 FUTURE DIRECTIONS**

3535       None.

**3536 SEE ALSO**

3537       Section 2.14 (on page 64)

**3538 CHANGE HISTORY**

3539       None.

**3540 NAME**

3541 times — write process times

**3542 SYNOPSIS**

3543 times

**3544 DESCRIPTION**

3545 Write the accumulated user and system times for the shell and for all of its child processes, in the  
3546 following POSIX locale format:

```
3547 "%dm%fs %dm%fs\n%dm%fs %dm%fs\n", <shell user minutes>,
3548 <shell user seconds>, <shell system minutes>,
3549 <shell system seconds>, <children user minutes>,
3550 <children user seconds>, <children system minutes>,
3551 <children system seconds>
```

3552 The four pairs of times shall correspond to the members of the <*sys/times.h*> **tms** structure  
3553 (defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 13, Headers) as  
3554 returned by *times()*: *tms\_utime*, *tms\_stime*, *tms\_cutime*, and *tms\_cstime*, respectively.

**3555 OPTIONS**

3556 None.

**3557 OPERANDS**

3558 None.

**3559 STDIN**

3560 None.

**3561 INPUT FILES**

3562 None.

**3563 ENVIRONMENT VARIABLES**

3564 None.

**3565 ASYNCHRONOUS EVENTS**

3566 None.

**3567 STDOUT**

3568 None.

**3569 STDERR**

3570 None.

**3571 OUTPUT FILES**

3572 None.

**3573 EXTENDED DESCRIPTION**

3574 None.

**3575 EXIT STATUS**

3576 Zero.

**3577 CONSEQUENCES OF ERRORS**

3578 None.

**3579 APPLICATION USAGE**

3580 None.

**3581 EXAMPLES**

```
3582 $ times
3583 0m0.43s 0m1.11s
3584 8m44.18s 1m43.23s
```

**3585 RATIONALE**

3586 The *times* special built-in from the Single UNIX Specification is now required for all conforming  
3587 shells.

**3588 FUTURE DIRECTIONS**

3589 None.

**3590 SEE ALSO**

3591 Section 2.14 (on page 64)

**3592 CHANGE HISTORY**

3593 None.

3594 **NAME**

3595 trap — trap signals

3596 **SYNOPSIS**3597 trap [*action condition ...*]3598 **DESCRIPTION**

3599 If *action* is ‘-’, the shell shall reset each *condition* to the default value. If *action* is null (" "), the  
3600 shell shall ignore each specified *condition* if it arises. Otherwise, the argument *action* shall be read  
3601 and executed by the shell when one of the corresponding conditions arises. The action of *trap*  
3602 shall override a previous action (either default action or one explicitly set). The value of "\$?"  
3603 after the *trap* action completes shall be the value it had before *trap* was invoked.

3604 The condition can be EXIT, 0 (equivalent to EXIT), or a signal specified using a symbolic name,  
3605 without the SIG prefix, as listed in the tables of signal names in the <signal.h> header defined in  
3606 the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 13, Headers; for example, HUP,  
3607 INT, QUIT, TERM. Implementations may permit names with the SIG prefix or ignore case in  
3608 signal names as an extension. Setting a trap for SIGKILL or SIGSTOP produces undefined  
3609 results.

3610 The environment in which the shell executes a *trap* on EXIT shall be identical to the environment  
3611 immediately after the last command executed before the *trap* on EXIT was taken.

3612 Each time *trap* is invoked, the *action* argument shall be processed in a manner equivalent to:

3613 eval *action*

3614 Signals that were ignored on entry to a non-interactive shell cannot be trapped or reset, although  
3615 no error need be reported when attempting to do so. An interactive shell may reset or catch  
3616 signals ignored on entry. Traps shall remain in place for a given shell until explicitly changed  
3617 with another *trap* command.

3618 When a subshell is entered, traps that are not being ignored are set to the default actions. This  
3619 does not imply that the *trap* command cannot be used within the subshell to set new traps.

3620 The *trap* command with no arguments shall write to standard output a list of commands  
3621 associated with each condition. The format shall be:

3622 "trap -- %s %s ... \n", <*action*>, <*condition*> ...

3623 The shell shall format the output, including the proper use of quoting, so that it is suitable for  
3624 reinput to the shell as commands that achieve the same trapping results. For example:

3625 save\_traps=\$(trap)  
3626 ...  
3627 eval "\$save\_traps"

3628 XSI-conformant systems also allow numeric signal numbers for the conditions corresponding to  
3629 the following signal names:

|          | <b>Signal Number</b> | <b>Signal Name</b> |
|----------|----------------------|--------------------|
| 3632 XSI | 1                    | SIGHUP             |
| 3633 XSI | 2                    | SIGINT             |
| 3634 XSI | 3                    | SIGQUIT            |
| 3635 XSI | 6                    | SIGABRT            |
| 3636 XSI | 9                    | SIGKILL            |
| 3637 XSI | 14                   | SIGALRM            |
| 3638 XSI | 15                   | SIGTERM            |

3639        The *trap* special built-in shall conform to the Base Definitions volume of IEEE Std 1003.1-2001,  
 3640        Section 12.2, Utility Syntax Guidelines.

#### 3641 **OPTIONS**

3642        None.

#### 3643 **OPERANDS**

3644        None.

#### 3645 **STDIN**

3646        None.

#### 3647 **INPUT FILES**

3648        None.

#### 3649 **ENVIRONMENT VARIABLES**

3650        None.

#### 3651 **ASYNCHRONOUS EVENTS**

3652        None.

#### 3653 **STDOUT**

3654        None.

#### 3655 **STDERR**

3656        None.

#### 3657 **OUTPUT FILES**

3658        None.

#### 3659 **EXTENDED DESCRIPTION**

3660        None.

#### 3661 **EXIT STATUS**

3662 XSI        If the trap name or number is invalid, a non-zero exit status shall be returned; otherwise, zero  
 3663 XSI        shall be returned. For both interactive and non-interactive shells, invalid signal names or  
 3664        numbers shall not be considered a syntax error and do not cause the shell to abort.

#### 3665 **CONSEQUENCES OF ERRORS**

3666        None.

3667 **APPLICATION USAGE**

3668 None.

3669 **EXAMPLES**

3670 Write out a list of all traps and actions:

3671 trap

3672 Set a trap so the *logout* utility in the directory referred to by the *HOME* environment variable  
3673 executes when the shell terminates:

3674 trap '\$HOME/logout' EXIT

3675 or:

3676 trap '\$HOME/logout' 0

3677 Unset traps on INT, QUIT, TERM, and EXIT:

3678 trap - INT QUIT TERM EXIT

3679 **RATIONALE**3680 Implementations may permit lowercase signal names as an extension. Implementations may  
3681 also accept the names with the SIG prefix; no known historical shell does so. The *trap* and *kill*  
3682 utilities in this volume of IEEE Std 1003.1-2001 are now consistent in their omission of the SIG  
3683 prefix for signal names. Some *kill* implementations do not allow the prefix, and *kill -l* lists the  
3684 signals without prefixes.3685 Trapping SIGKILL or SIGSTOP is syntactically accepted by some historical implementations, but  
3686 it has no effect. Portable POSIX applications cannot attempt to trap these signals.3687 The output format is not historical practice. Since the output of historical *trap* commands is not  
3688 portable (because numeric signal values are not portable) and had to change to become so, an  
3689 opportunity was taken to format the output in a way that a shell script could use to save and  
3690 then later reuse a trap if it wanted.3691 The KornShell uses an **ERR** trap that is triggered whenever *set -e* would cause an exit. This is  
3692 allowable as an extension, but was not mandated, as other shells have not used it.3693 The text about the environment for the EXIT trap invalidates the behavior of some historical  
3694 versions of interactive shells which, for example, close the standard input before executing a  
3695 trap on 0. For example, in some historical interactive shell sessions the following trap on 0 would  
3696 always print "--":

3697 trap 'read foo; echo "-\$foo-' 0

3698 **FUTURE DIRECTIONS**

3699 None.

3700 **SEE ALSO**

3701 Section 2.14 (on page 64)

3702 **CHANGE HISTORY**3703 **Issue 6**3704 XSI-conforming implementations provide the mapping of signal names to numbers given above  
3705 (previously this had been marked obsolescent). Other implementations need not provide this  
3706 optional mapping.

**3707 NAME**

3708        unset — unset values and attributes of variables and functions

**3709 SYNOPSIS**

3710        `unset [-fv] name ...`

**3711 DESCRIPTION**

3712        Each variable or function specified by *name* shall be unset.

3713        If **-v** is specified, *name* refers to a variable name and the shell shall unset it and remove it from  
3714        the environment. Read-only variables cannot be unset.

3715        If **-f** is specified, *name* refers to a function and the shell shall unset the function definition.

3716        If neither **-f** nor **-v** is specified, *name* refers to a variable; if a variable by that name does not  
3717        exist, it is unspecified whether a function by that name, if any, shall be unset.

3718        Unsetting a variable or function that was not previously set shall not be considered an error and  
3719        does not cause the shell to abort.

3720        The *unset* special built-in shall support the Base Definitions volume of IEEE Std 1003.1-2001,  
3721        Section 12.2, Utility Syntax Guidelines.

3722        Note that:

3723        **VARIABLE**=

3724        is not equivalent to an *unset* of **VARIABLE**; in the example, **VARIABLE** is set to " ". Also, the  
3725        variables that can be *unset* should not be misinterpreted to include the special parameters (see  
3726        Section 2.5.2 (on page 34)).

**3727 OPTIONS**

3728        None.

**3729 OPERANDS**

3730        None.

**3731 STDIN**

3732        None.

**3733 INPUT FILES**

3734        None.

**3735 ENVIRONMENT VARIABLES**

3736        None.

**3737 ASYNCHRONOUS EVENTS**

3738        None.

**3739 STDOUT**

3740        None.

**3741 STDERR**

3742        None.

**3743 OUTPUT FILES**

3744        None.

**3745 EXTENDED DESCRIPTION**

3746        None.

**3747 EXIT STATUS**

- 3748        0 All *name* operands were successfully unset.  
3749        >0 At least one *name* could not be unset.

**3750 CONSEQUENCES OF ERRORS**

- 3751        None.

**3752 APPLICATION USAGE**

- 3753        None.

**3754 EXAMPLES**

- 3755        Unset *VISUAL* variable:

3756        `unset -v VISUAL`

- 3757        Unset the functions **foo** and **bar**:

3758        `unset -f foo bar`

**3759 RATIONALE**

3760        Consideration was given to omitting the **-f** option in favor of an *unfunction* utility, but the  
3761        standard developers decided to retain historical practice.

3762        The **-v** option was introduced because System V historically used one name space for both  
3763        variables and functions. When *unset* is used without options, System V historically unset either a  
3764        function or a variable, and there was no confusion about which one was intended. A portable  
3765        POSIX application can use *unset* without an option to unset a variable, but not a function; the **-f**  
3766        option must be used.

**3767 FUTURE DIRECTIONS**

- 3768        None.

**3769 SEE ALSO**

- 3770        Section 2.14 (on page 64)

**3771 CHANGE HISTORY**

- 3772        None.



# Batch Environment Services

3774 BE This chapter describes the services and utilities that shall be implemented on all systems that  
 3775 claim conformance to the Batch Environment option. This functionality is dependent on support  
 3776 of this option (and the rest of this section is not further shaded for this option).

## 3.1 General Concepts

### 3.1.1 Batch Client-Server Interaction

3779 Batch jobs are created and managed by batch servers. A batch client interacts with a batch server  
 3780 to access batch services on behalf of the user. In order to use batch services, a user must have  
 3781 access to a batch client.

3782 A batch server is a computational entity, such as a daemon process, that provides batch services.  
 3783 Batch servers route, queue, modify, and execute batch jobs on behalf of batch clients.

3784 The batch utilities described in this volume of IEEE Std 1003.1-2001 (and listed in Table 3-1) are  
 3785 clients of batch services; they allow users to perform actions on the job such as creating,  
 3786 modifying, and deleting batch jobs from a shell command line. Although these batch utilities  
 3787 may be said to accomplish certain services, they actually obtain services on behalf of a user by  
 3788 means of requests to batch servers.

3789 **Table 3-1** Batch Utilities

|               |               |                |              |
|---------------|---------------|----------------|--------------|
| <i>qalter</i> | <i>qmove</i>  | <i>qrsl</i>    | <i>qstat</i> |
| <i>qdel</i>   | <i>qmsg</i>   | <i>qselect</i> | <i>qsub</i>  |
| <i>qhold</i>  | <i>qrerun</i> | <i>qsig</i>    |              |

3793 Client-server interaction takes place by means of the batch requests defined in this chapter.  
 3794 Because direct access to batch jobs and queues is limited to batch servers, clients and servers of  
 3795 different implementations can interoperate, since dependencies on private structures for batch  
 3796 jobs and queues are limited to batch servers. Also, batch servers may be clients of other batch  
 3797 servers.

### 3.1.2 Batch Queues

3799 Two types of batch queue are described: routing queues and execution queues. When a batch job  
 3800 is placed in a routing queue, it is a candidate for routing. A batch job is removed from routing  
 3801 queues under the following conditions:

- 3802 • The batch job has been routed to another queue.
- 3803 • The batch job has been deleted from the batch queue.
- 3804 • The batch job has been aborted.

3805 When a batch job is placed in an execution queue, it is a candidate for execution.

3806 A batch job is removed from an execution queue under the following conditions:

- 3807 • The batch job has been executed and exited.

- 3808           • The batch job has been aborted.  
3809           • The batch job has been deleted from the batch queue.  
3810           • The batch job has been moved to another queue.

3811 Access to a batch queue is limited to the batch server that manages the batch queue. Clients  
3812 never access a batch queue or a batch job directly, either to read or write information; all client  
3813 access to batch queues or jobs takes place through batch servers.

### 3814 **3.1.3 Batch Job Creation**

3815 When a batch server creates a batch job on behalf of a client, it shall assign a batch job identifier  
3816 to the job. A batch job identifier consists of both a sequence number that is unique among the  
3817 sequence numbers issued by that server and the name of the server. Since the batch server name  
3818 is unique within a name space, the job identifier is likewise unique within the name space.

3819 The batch server that creates a batch job shall return the batch server-assigned job identifier to  
3820 the client that requested the job creation. If the batch server routes or moves the job to another  
3821 server, it sends the job identifier with the job. Once assigned, the job identifier of a batch job shall  
3822 never change.

### 3823 **3.1.4 Batch Job Tracking**

3824 Since a batch job may be moved after creation, the batch server name component of the job  
3825 identifier need not indicate the location of the job. An implementation may provide a batch job  
3826 tracking mechanism, in which case the user generally does not need to know the location of the  
3827 job. However, an implementation need not provide a batch job tracking mechanism, in which  
3828 case the user must find routed jobs by probing the possible destinations.

### 3829 **3.1.5 Batch Job Routing**

3830 To route a batch job, a batch server either moves the job to some other queue that is managed by  
3831 the batch server, or requests that some other batch server accept the job.

3832 Each routing queue has one or more queues to which it can route batch jobs. The batch server  
3833 administrator creates routing queues.

3834 A batch server may route a batch job from a routing queue to another routing queue. Batch  
3835 servers shall prevent or otherwise handle cases of circular routing paths. As a deferred service, a  
3836 batch server routes jobs from the routing queues that it manages. The algorithm by which a  
3837 batch server selects a batch queue to which to route a batch job is implementation-defined.

3838 A batch job need not be eligible for routing to all the batch queues fed by the routing queue from  
3839 which it is routed. A batch server that has been asked to accept the job may reject the request if  
3840 the job requires resources that are unavailable to that batch server, or if the client is not  
3841 authorized to access the batch server.

3842 Batch servers may route high-priority jobs before low-priority jobs, but, on other than  
3843 overloaded systems, the effect may be imperceptible to the user. If all the batch servers fed by a  
3844 routing queue reject requests to accept the job for reasons that are permanent, the batch server  
3845 that manages the job shall abort the job. If all or some rejections are temporary, the batch server  
3846 should try to route the job again at some later point.

3847 The reasons for rejecting a batch job are implementation-defined. The reasons for which the  
3848 routing should be retried later and the reasons for which the job should be aborted are also  
3849 implementation-defined.

**3850 3.1.6 Batch Job Execution**

3851 To execute a batch job is to create a session leader (a process) that runs the shell program  
3852 indicated by the *Shell\_Path* attribute of the job. The script shall be passed to the program as its  
3853 standard input. An implementation may pass the script to the program by other  
3854 implementation-defined means. At the time a batch job begins execution, it is defined to enter  
3855 the RUNNING state. The primary program that is executed by a batch job is typically, though  
3856 not necessarily, a shell program.

3857 A batch server shall execute eligible jobs as a deferred service—no client request is necessary  
3858 once the batch job is created and eligible. However, the attributes of a batch job, such as the job  
3859 hold type, may render the job ineligible. A batch server shall scan the execution queues that it  
3860 manages for jobs that are eligible for execution. The algorithm by which the batch server selects  
3861 eligible jobs for execution is implementation-defined.

3862 As part of creating the process for the batch job, the batch server shall open the standard output  
3863 and standard error streams of the session.

3864 The attributes of a batch job may indicate that the batch server executing the job shall send mail  
3865 to a list of users at the time it begins execution of the job.

**3866 3.1.7 Batch Job Exit**

3867 When the session leader of an executing job terminates, the job exits. As part of exiting a batch  
3868 job, the batch server that manages the job shall remove the job from the batch queue in which it  
3869 resides. The server shall transfer output files of the job to a location described by the attributes of  
3870 the job.

3871 The attributes of a batch job may indicate that the batch server managing the job shall send mail  
3872 to a list of users at the time the job exits.

**3873 3.1.8 Batch Job Abort**

3874 A batch server shall abort jobs for which a required deferred service cannot be performed. The  
3875 attributes of a batch job may indicate that the batch server that aborts the job shall send mail to a  
3876 list of users at the time it aborts the job.

**3877 3.1.9 Batch Authorization**

3878 Clients, such as the batch environment utilities (marked BE), access batch services by means of  
3879 requests to one or more batch servers. To acquire the services of any given batch server, the user  
3880 identifier under which the client runs must be authorized to use that batch server.

3881 The user with an associated user name that creates a batch job shall own the job and can perform  
3882 actions such as read, modify, delete, and move.

3883 A user identifier of the same value at a different host need not be the same user. For example,  
3884 user name *smith* at host **alpha** may or may not represent the same person as user name *smith* at  
3885 host **beta**. Likewise, the same person may have access to different user names on different hosts.

3886 An implementation may optionally provide an authorization mechanism that permits one user  
3887 name to access jobs under another user name.

3888 A process on a client host may be authorized to run processes under multiple user names at a  
3889 batch server host. Where appropriate, the utilities defined in this volume of IEEE Std 1003.1-2001  
3890 provide a means for a user to choose from among such user names when creating or modifying a  
3891 batch job.

3892 **3.1.10 Batch Administration**

3893 The processing of a batch job by a batch server is affected by the attributes of the job. The  
3894 processing of a batch job may also be affected by the attributes of the batch queue in which the  
3895 job resides and by the status of the batch server that manages the job. See also the Base  
3896 Definitions volume of IEEE Std 1003.1-2001, Chapter 3, Definitions for batch definitions.

3897 **3.1.11 Batch Notification**

3898 Whereas batch servers are persistent entities, clients are often transient. For example, the *qsub*  
3899 utility creates a batch job and exits. For this reason, batch servers notify users of batch job events  
3900 by sending mail to the user that owns the job, or to other designated users.

3901 **3.2 Batch Services**

3902 The presence of Batch Environment option services is indicated by the configuration variable  
3903 `POSIX2_PBS`. A conforming batch server provides services as defined in this section.

3904 A batch server shall provide batch services in two ways:

- 3905 1. The batch server provides a service at the request of a client.  
3906 2. The batch server provides a deferred service as a result of a change in conditions  
3907 monitored by the batch server.

3908 If a batch server cannot complete a request, it shall reject the request. If a batch server cannot  
3909 complete a deferred service for a batch job, the batch server shall abort the batch job. Table 3-2  
3910 (on page 105) is a summary of environment variables that shall be supported by an  
3911 implementation of the batch server and utilities.

3912

**Table 3-2** Environment Variable Summary

| Variable               | Description                                                          |
|------------------------|----------------------------------------------------------------------|
| <i>PBS_DPREFIX</i>     | Defines the directive prefix (see <i>qsub</i> )                      |
| <i>PBS_ENVIRONMENT</i> | Batch Job is batch or interactive (see Section 3.2.2.1)              |
| <i>PBS_JOBID</i>       | The <i>job_identifier</i> attribute of job (see Section 3.2.3.8)     |
| <i>PBS_JOBNAME</i>     | The <i>job_name</i> attribute of job (see Section 3.2.3.8)           |
| <i>PBS_O_HOME</i>      | Defines the <i>HOME</i> of the batch client (see <i>qsub</i> )       |
| <i>PBS_O_HOST</i>      | Defines the host name of the batch client (see <i>qsub</i> )         |
| <i>PBS_O_LANG</i>      | Defines the <i>LANG</i> of the batch client (see <i>qsub</i> )       |
| <i>PBS_O_LOGNAME</i>   | Defines the <i>LOGNAME</i> of the batch client (see <i>qsub</i> )    |
| <i>PBS_O_MAIL</i>      | Defines the <i>MAIL</i> of the batch client (see <i>qsub</i> )       |
| <i>PBS_O_PATH</i>      | Defines the <i>PATH</i> of the batch client (see <i>qsub</i> )       |
| <i>PBS_O_QUEUE</i>     | Defines the submit queue of the batch client (see <i>qsub</i> )      |
| <i>PBS_O_SHELL</i>     | Defines the <i>SHELL</i> of the batch client (see <i>qsub</i> )      |
| <i>PBS_O_TZ</i>        | Defines the <i>TZ</i> of the batch client (see <i>qsub</i> )         |
| <i>PBS_O_WORKDIR</i>   | Defines the working directory of the batch client (see <i>qsub</i> ) |
| <i>PBS_QUEUE</i>       | Defines the initial execution queue (see Section 3.2.2.1)            |

3929 **3.2.1 Batch Job States**

3930 A batch job shall always be in one of the following states: QUEUED, RUNNING, HELD,  
 3931 WAITING, EXITING, or TRANSITING. The state of a batch job determines the types of requests  
 3932 that the batch server that manages the batch job can accept for the batch job. A batch server shall  
 3933 change the state of a batch job either in response to service requests from clients or as a result of  
 3934 deferred services, such as job execution or job routing.

3935 A batch job that is in the QUEUED state resides in a queue but is still pending either execution or  
 3936 routing, depending on the queue type.

3937 A batch server that queues a batch job in a routing queue shall put the batch job in the QUEUED  
 3938 state. A batch server that puts a batch job in an execution queue, but has not yet executed the  
 3939 batch job, shall put the batch job in the QUEUED state. A batch job that resides in an execution  
 3940 queue and is executing is defined to be in the RUNNING state. While a batch job is in the  
 3941 RUNNING state, a session leader is associated with the batch job.

3942 A batch job that resides in an execution queue, but is ineligible to run because of a hold attribute,  
 3943 is defined to be in the HELD state.

3944 A batch job that is not held, but must wait until a future date and time before executing, is  
 3945 defined to be in the WAITING state.

3946 When the session leader associated with a running job exits, the batch job shall be placed in the  
 3947 EXITING state.

3948 A batch job for which the session leader has terminated is defined to be in the EXITING state,  
 3949 and the batch server that manages such a batch job cannot accept job modification requests that  
 3950 affect the batch job. While a batch job is in the EXITING state, the batch server that manages the  
 3951 batch job is staging output files and notifying clients of job completion. Once a batch job has  
 3952 exited, it no longer exists as an object managed by a batch server.

3953 A batch job that is being moved from a routing queue to another queue is defined to be in the  
 3954 TRANSITING state.

3955 When a batch job in a routing queue has been selected to be moved to a new destination, then  
3956 the batch job shall be in either the QUEUED state or the TRANSITING state, depending on the  
3957 batch server implementation.

3958 Batch jobs with either an *Execution\_Time* attribute value set in the future or a *Hold\_Types* attribute  
3959 of value not equal to NO\_HOLD, or both, may be routed or held in the routing queue. The  
3960 treatment of jobs with the *Execution\_Time* or *Hold\_Types* attributes in a routing queue is  
3961 implementation-defined.

3962 When a batch job in a routing queue has not been selected to be moved to a new destination and  
3963 the batch job has a *Hold\_Types* attribute value of other than NO\_HOLD, then the job should be in  
3964 the HELD state.

3965 **Note:** The effect of a hold upon a batch job in a routing queue is implementation-defined. The  
3966 implementation should use the state that matches whether the batch job can route with a hold  
3967 or not.

3968 When a batch job in a routing queue has not been selected to be moved to a new destination and  
3969 the batch job has:

- A *Hold\_Types* attribute value of NO\_HOLD
- An *Execution\_Time* attribute in the past

3972 then the batch job shall be in the QUEUED state.

3973 When a batch job in a routing queue has not been selected to be moved to a new destination and  
3974 the batch job has:

- A *Hold\_Types* attribute value of NO\_HOLD
- An *Execution\_Time* attribute in the future

3977 then the batch job may be in the WAITING state.

3978 **Note:** The effect of a future execution time upon a batch job in a routing queue is implementation-  
3979 defined. The implementation should use the state that matches whether the batch job can route  
3980 with a hold or not.

3981 Table 3-3 (on page 107) describes the next state of a batch job, given the current state of the batch  
3982 job and the type of request. Table 3-4 (on page 108) describes the response of a batch server to a  
3983 request, given the current state of the batch job and the type of request.

### 3984 3.2.2 Deferred Batch Services

3985 This section describes the deferred services performed by batch servers: job execution, job  
3986 routing, job exit, job abort, and the rerunning of jobs after a restart.

#### 3987 3.2.2.1 Batch Job Execution

3988 To execute a batch job is to create a session leader (a process) that runs the shell program  
3989 indicated by the *Shell\_Path\_List* attribute of the batch job. The script is passed to the program as  
3990 its standard input. An implementation may pass the script to the program by other  
3991 implementation-defined means. At the time a batch job begins execution, it is defined to enter  
3992 the RUNNING state.

3993

**Table 3-3** Next State Table

| Request Type               | Current State |   |     |       |   |   |   |
|----------------------------|---------------|---|-----|-------|---|---|---|
|                            | X             | Q | R   | H     | W | E | T |
| Queue Batch Job Request    | Q             | e | e   | e     | e | e | e |
| Modify Batch Job Request   | e             | Q | R   | H     | W | e | T |
| Delete Batch Job Request   | e             | X | E   | X     | X | E | X |
| Batch Job Message Request  | e             | Q | R   | H     | W | E | T |
| Rerun Batch Job Request    | e             | e | Q   | e     | e | e | e |
| Signal Batch Job Request   | e             | e | R   | H     | W | e | e |
| Batch Job Status Request   | e             | Q | R   | H     | W | E | T |
| Batch Queue Status Request | X             | Q | R   | H     | W | E | T |
| Server Status Request      | X             | Q | R   | H     | W | E | T |
| Select Batch Jobs Request  | X             | Q | R   | H     | W | E | T |
| Move Batch Job Request     | e             | Q | R   | H     | W | e | T |
| Hold Batch Job Request     | e             | H | R/H | H     | H | e | T |
| Release Batch Job Request  | e             | Q | R   | Q/W/H | W | e | T |
| Server Shutdown Request    | X             | Q | Q   | H     | W | E | T |
| Locate Batch Job Request   | e             | Q | R   | H     | W | E | T |

4011      **Legend**

4012      X    Nonexistent

4013      Q    QUEUED

4014      R    RUNNING

4015      H    HELD

4016      W    WAITING

4017      E    EXITING

4018      T    TRANSITING

4019      e    Error

4020      A batch server that has an execution queue containing jobs is said to own the queue and manage  
 4021      the batch jobs in that queue. A batch server that has been started shall execute the batch jobs in  
 4022      the execution queues owned by the batch server. The batch server shall schedule for execution  
 4023      those jobs in the execution queues that are in the QUEUED state. The algorithm for scheduling  
 4024      jobs is implementation-defined.

4025      A batch server that executes a batch job shall create, in the environment of the session leader of  
 4026      the batch job, an environment variable named *PBS\_ENVIRONMENT*, the value of which is the  
 4027      string PBS\_BATCH encoded in the portable character set.

4028      A batch server that executes a batch job shall create, in the environment of the session leader of  
 4029      the batch job, an environment variable named *PBS\_QUEUE*, the value of which is the name of  
 4030      the execution queue of the batch job encoded in the portable character set.

4031      To rerun a batch job is to requeue a batch job that is currently executing and then kill the session  
 4032      leader of the executing job by sending a SIGKILL prior to completion; see Section 3.2.3.11 (on  
 4033      page 120). A batch server that reruns a batch job shall append the standard output and standard  
 4034      error files of the batch job to the corresponding files of the previous execution, if they exist, with  
 4035      appropriate annotation. If either file does not exist, that file shall be created as in normal  
 4036      execution.

4037

**Table 3-4** Results/Output Table

| Request Type               | Current State |   |   |   |   |   |   |
|----------------------------|---------------|---|---|---|---|---|---|
|                            | X             | Q | R | H | W | E | T |
| Queue Batch Job Request    | O             | e | e | e | e | e | e |
| Modify Batch Job Request   | e             | O | e | O | O | e | e |
| Delete Batch Job Request   | e             | O | O | O | O | e | O |
| Batch Job Message Request  | e             | e | O | e | e | e | e |
| Rerun Batch Job Request    | e             | e | O | e | e | e | e |
| Signal Batch Job Request   | e             | e | O | e | e | e | e |
| Batch Job Status Request   | e             | O | O | O | O | O | O |
| Batch Queue Status Request | O             | O | O | O | O | O | O |
| Server Status Request      | O             | O | O | O | O | O | O |
| Select Batch Job Request   | e             | O | O | O | O | O | O |
| Move Batch Job Request     | e             | O | O | O | O | e | e |
| Hold Batch Job Request     | e             | O | O | O | O | e | e |
| Release Batch Job Request  | e             | O | e | O | O | e | e |
| Server Shutdown Request    | O             | O | e | O | O | e | e |
| Locate Batch Job Request   | e             | O | O | O | O | O | O |

4055      **Legend**

4056      O    OK

4057      e    Error message

4058      The execution of a batch job by a batch server shall be controlled by job, queue, and server  
4059      attributes, as defined in this section.4060      **Account\_Name Attribute**4061      Batch accounting is an optional feature of batch servers. If a batch server implements  
4062      accounting, the statements in this section apply and the configuration variable  
4063      POSIX2\_PBS\_ACCOUNTING shall be set to 1.4064      A batch server that executes a batch job shall charge the account named in the *Account\_Name*  
4065      attribute of the batch job for resources consumed by the batch job.4066      If the *Account\_Name* attribute of the batch job is absent from the batch job attribute list or is  
4067      altered while the batch job is in execution, the batch server action is implementation-defined.4068      **Checkpoint Attribute**4069      Batch checkpointing is an optional feature of batch servers. If a batch server implements  
4070      checkpointing, the statements in this section apply and the configuration variable  
4071      POSIX2\_PBS\_CHECKPOINT shall be set to 1.4072      There are two attributes associated with the checkpointing feature: *Checkpoint* and  
4073      *Minimum\_Cpu\_Interval*. *Checkpoint* is a batch job attribute, while *Minimum\_Cpu\_Interval* is a  
4074      queue attribute. An implementation that does not support checkpointing shall support the  
4075      *Checkpoint* job attribute to the extent that the batch server shall maintain and pass this attribute  
4076      to other servers.4077      The behavior of a batch server that executes a batch job for which the value of the *Checkpoint*  
4078      attribute is CHECKPOINT\_UNSPECIFIED is implementation-defined. A batch server that  
4079      executes a batch job for which the value of the *Checkpoint* attribute is NO\_CHECKPOINT shall

- 4080       not checkpoint the batch job.
- 4081       A batch server that executes a batch job for which the value of the *CHECKPOINT* attribute is  
4082       CHECKPOINT\_AT\_SHUTDOWN shall checkpoint the batch job only when the batch server  
4083       accepts a request to shut down during the time when the batch job is in the RUNNING state.
- 4084       A batch server that executes a batch job for which the value of the *CHECKPOINT* attribute is  
4085       CHECKPOINT\_AT\_MIN\_CPU\_INTERVAL shall checkpoint the batch job at the interval  
4086       specified by the *Minimum\_Cpu\_Interval* attribute of the queue for which the batch job has been  
4087       selected. The *Minimum\_Cpu\_Interval* attribute shall be specified in units of CPU minutes.
- 4088       A batch server that executes a batch job for which the value of the *CHECKPOINT* attribute is an  
4089       unsigned integer shall checkpoint the batch job at an interval that is the value of either the  
4090       *CHECKPOINT* attribute, or the *Minimum\_Cpu\_Interval* attribute of the queue for which the batch job  
4091       has been selected, whichever is greater. Both intervals shall be in units of CPU minutes. When  
4092       the *Minimum\_Cpu\_Interval* attribute is greater than the *CHECKPOINT* attribute, the batch job shall  
4093       write a warning message to the standard error stream of the batch job.
- 4094       **Error\_Path Attribute**
- 4095       The *Error\_Path* attribute of a running job cannot be changed by a *Modify Batch Job Request*. When  
4096       the *Join\_Path* attribute of the batch job is set to the value FALSE and the *Keep\_Files* attribute of  
4097       the batch job does not contain the value KEEP\_STD\_ERROR, a batch server that executes a batch  
4098       job shall perform one of the following actions:
- 4099       • Set the standard error stream of the session leader of the batch job to the path described by  
4100        the value of the *Error\_Path* attribute of the batch job.
  - 4101       • Buffer the standard error of the session leader of the batch job until completion of the batch  
4102        job, and when the batch job exits return the contents to the destination described by the value  
4103        of the *Error\_Path* attribute of the batch job.
- 4104       Applications shall not rely on having access to the standard error of a batch job prior to the  
4105       completion of the batch job.
- 4106       When the *Error\_Path* attribute does not specify a host name, then the batch server shall retain the  
4107       standard error of the batch job on the host of execution.
- 4108       When the *Error\_Path* attribute does specify a host name and the *Keep\_Files* attribute does not  
4109       contain the value KEEP\_STD\_ERROR, then the final destination of the standard error of the  
4110       batch job shall be on the host whose host name is specified.
- 4111       If the path indicated by the value of the *Error\_Path* attribute of the batch job is a relative path, the  
4112       batch server shall expand the path relative to the home directory of the user on the host to which  
4113       the file is being returned.
- 4114       When the batch server buffers the standard error of the batch job and the file cannot be opened  
4115       for write upon completion of the batch job, then the server shall place the standard error in an  
4116       implementation-defined location and notify the user of the location via mail. It shall be possible  
4117       for the user to process this mail using the *mailx* utility.
- 4118       If a batch server that does not buffer the standard error cannot open the standard error path of  
4119       the batch job for write access, then the batch server shall abort the batch job.

4120           **Execution\_Time Attribute**

4121           A batch server shall not execute a batch job before the time represented by the value of the  
 4122           *Execution\_Time* attribute of the batch job. The *Execution\_Time* attribute is defined in seconds since  
 4123           the Epoch.

4124           **Hold\_Types Attribute**

4125           A batch server shall support the following hold types:

- 4126           s    Can be set or released by a user with at least a privilege level of batch administrator  
                4127           (SYSTEM).
- 4128           o    Can be set or released by a user with at least a privilege level of batch operator  
                4129           (OPERATOR).
- 4130           u    Can be set or released by the user with at least a privilege level of user, where the user is  
                4131           defined in the *Job\_Owner* attribute (USER).
- 4132           n    Indicates that none of the *Hold\_Types* attributes are set (NO\_HOLD).

4133           An implementation may define other hold types. Any additional hold types, how they are  
 4134           specified, their internal representation, their behavior, and how they affect the behavior of other  
 4135           utilities are implementation-defined.

4136           The value of the *Hold\_Types* attribute shall be the union of the valid hold types ('s', 'o', 'u',  
 4137           and any implementation-defined hold types), or 'n'.

4138           A batch server shall not execute a batch job if the *Hold\_Types* attribute of the batch job has a  
 4139           value other than NO\_HOLD. If the *Hold\_Types* attribute of the batch job has a value other than  
 4140           NO\_HOLD, the batch job shall be in the HELD state.

4141           **Job\_Owner Attribute**

4142           The *Job\_Owner* attribute consists of a pair of user name and host name values of the form:

4143            username@hostname

4144           A batch server that accepts a *Queue Batch Job Request* shall set the *Job\_Owner* attribute to a string  
 4145           that is the *username@hostname* of the user who submitted the job.

4146           **Join\_Path Attribute**

4147           A batch server that executes a batch job for which the value of the *Join\_Path* attribute is TRUE  
 4148           shall ignore the value of the *Error\_Path* attribute and merge the standard error of the batch job  
 4149           with the standard output of the batch job.

4150           **Keep\_Files Attribute**

4151           A batch server that executes a batch job for which the value of the *Keep\_Files* attribute includes  
 4152           the value KEEP\_STD\_OUTPUT shall retain the standard output of the batch job on the host  
 4153           where execution occurs. The standard output shall be retained in the home directory of the user  
 4154           under whose user ID the batch job is executed and the filename shall be the default filename for  
 4155           the standard output as defined under the -o option of the *qsub* utility. The *Output\_Path* attribute  
 4156           is not modified.

4157           A batch server that executes a batch job for which the value of the *Keep\_Files* attribute includes  
 4158           the value KEEP\_STD\_ERROR shall retain the standard error of the batch job on the host where  
 4159           execution occurs. The standard error shall be retained in the home directory of the user under  
 4160           whose user ID the batch job is executed and the filename shall be the default filename for

4161 standard error as defined under the `-e` option of the *qsub* utility. The *Error\_Path* attribute is not  
4162 modified.

4163 A batch server that executes a batch job for which the value of the *Keep\_Files* attribute includes  
4164 values other than *KEEP\_STD\_OUTPUT* and *KEEP\_STD\_ERROR* shall retain these other files on  
4165 the host where execution occurs. These files (with implementation-defined names) shall be  
4166 retained in the home directory of the user under whose user identifier the batch job is executed.

#### 4167 **Mail\_Points and Mail\_Users Attributes**

4168 A batch server that executes a batch job for which one of the values of the *Mail\_Points* attribute is  
4169 the value *MAIL\_AT\_BEGINNING* shall send a mail message to each user account listed in the  
4170 *Mail\_Users* attribute of the batch job.

4171 The mail message shall contain at least the batch job identifier, queue, and server at which the  
4172 batch job currently resides, and the *Job\_Owner* attribute.

#### 4173 **Output\_Path Attribute**

4174 The *Output\_Path* attribute of a running job cannot be changed by a *Modify Batch Job Request*.  
4175 When the *Keep\_Files* attribute of the batch job does not contain the value *KEEP\_STD\_OUTPUT*, a  
4176 batch server that executes a batch job shall either:

- 4177 • Set the standard output stream of the session leader of the batch job to the destination  
4178 described by the value of the *Output\_Path* attribute of the batch job.  
  
4179 or:  
  
4180 • Buffer the standard output of the session leader of the batch job until completion of the batch  
4181 job, and when the batch job exits return the contents to the destination described by the value  
4182 of the *Output\_Path* attribute of the batch job.

4183 When the *Output\_Path* attribute does not specify a host name, then the batch server shall retain  
4184 the standard output of the batch job on the host of execution.

4185 When the *Keep\_Files* attribute does not contain the value *KEEP\_STD\_OUTPUT* and the  
4186 *Output\_Path* attribute does specify a host name, then the final destination of the standard output  
4187 of the batch job shall be on the host specified.

4188 If the path specified in the *Output\_Path* attribute of the batch job is a relative path, the batch  
4189 server shall expand the path relative to the home directory of the user on the host to which the  
4190 file is being returned.

4191 Whether or not the batch server buffers the standard output of the batch job until completion of  
4192 the batch job is implementation-defined. Applications shall not rely on having access to the  
4193 standard output of a batch job prior to the completion of the batch job.

4194 When the batch server does buffer the standard output of the batch job and the file cannot be  
4195 opened for write upon completion of the batch job, then the batch server shall place the standard  
4196 output in an implementation-defined location and notify the user of the location via mail. It shall  
4197 be possible for the user to process this mail using the *mailx* utility.

4198 If a batch server that does not buffer the standard output cannot open the standard output path  
4199 of the batch job for write access, then the batch server shall abort the batch job.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4200 | <b>Priority Attribute</b>                                                                                                                                                                                                                                                                                                                                                                                                          |
| 4201 | A batch server implementation may choose to preferentially execute a batch job based on the <i>Priority</i> attribute. The interpretation of the batch job <i>Priority</i> attribute by a batch server is implementation-defined. If an implementation uses the <i>Priority</i> attribute, it shall interpret larger values of the <i>Priority</i> attribute to mean the batch job shall be preferentially selected for execution. |
| 4205 | <b>Rerunable Attribute</b>                                                                                                                                                                                                                                                                                                                                                                                                         |
| 4206 | A batch job that began execution but did not complete, because the batch server either shut down or terminated abnormally, shall be requeued if the <i>Rerunable</i> attribute of the batch job has the value TRUE.                                                                                                                                                                                                                |
| 4209 | If a batch job, which was requeued after beginning execution but prior to completion, has a valid checkpoint file and the batch server supports checkpointing, then the batch job shall be restarted from the last valid checkpoint.                                                                                                                                                                                               |
| 4212 | If the batch job cannot be restarted from a checkpoint, then when a batch job has a <i>Rerunable</i> attribute value of TRUE and was requeued after beginning execution but prior to completion, the batch server shall place the batch job into execution at the beginning of the job.                                                                                                                                            |
| 4215 | When a batch job has a <i>Rerunable</i> attribute value other than TRUE and was requeued after beginning execution but prior to completion, and the batch job cannot be restarted from a checkpoint, then the batch server shall abort the batch job.                                                                                                                                                                              |
| 4218 | <b>Resource_List Attribute</b>                                                                                                                                                                                                                                                                                                                                                                                                     |
| 4219 | A batch server that executes a batch job shall establish the resource limits of the session leader of the batch job according to the values of the <i>Resource_List</i> attribute of the batch job. Resource limits shall be enforced by an implementation-defined method.                                                                                                                                                         |
| 4222 | <b>Shell_Path_List Attribute</b>                                                                                                                                                                                                                                                                                                                                                                                                   |
| 4223 | The <i>Shell_Path_List</i> job attribute consists of a list of pairs of pathname and host name values. The host name component can be omitted, in which case the pathname serves as the default pathname when a batch server cannot find the name of the host on which it is running in the list.                                                                                                                                  |
| 4226 | A batch server that executes a batch job shall select, from the value of the <i>Shell_Path_List</i> attribute of the batch job, a pathname where the shell to execute the batch job shall be found. The batch server shall select the pathname, in order of preference, according to the following methods:                                                                                                                        |
| 4230 | <ul style="list-style-type: none"><li>• Select the pathname that contains the name of the host on which the batch server is running.</li></ul>                                                                                                                                                                                                                                                                                     |
| 4231 | <ul style="list-style-type: none"><li>• Select the pathname for which the host name has been omitted.</li></ul>                                                                                                                                                                                                                                                                                                                    |
| 4232 | <ul style="list-style-type: none"><li>• Select the pathname for the login shell of the user under which the batch job is to execute.</li></ul>                                                                                                                                                                                                                                                                                     |
| 4233 | If the shell path value selected is an invalid pathname, the batch server shall abort the batch job.                                                                                                                                                                                                                                                                                                                               |
| 4234 | If the value of the selected pathname from the <i>Shell_Path_List</i> attribute of the batch job represents a partial path, the batch server shall expand the path relative to a path that is implementation-defined.                                                                                                                                                                                                              |
| 4237 | The batch server that executes the batch job shall execute the program that was selected from the <i>Shell_Path_List</i> attribute of the batch job. The batch server shall pass the path to the script of the batch job as the first argument to the shell program.                                                                                                                                                               |

4240      **User\_List Attribute**

4241      The *User\_List* job attribute consists of a list of pairs of user name and host name values. The host  
 4242      name component can be omitted, in which case the user name serves as a default when a batch  
 4243      server cannot find the name of the host on which it is running in the list.

4244      A batch server that executes a batch job shall select, from the value of the *User\_List* attribute of  
 4245      the batch job, a user name under which to create the session leader. The server shall select the  
 4246      user name, in order of preference, according to the following methods:

- 4247      • Select the user name of a value that contains the name of the host on which the batch server  
   4248      executes.
- 4249      • Select the user name of a value for which the host name has been omitted.
- 4250      • Select the user name from the *Job\_Owner* attribute of the batch job.

4251      **Variable\_List Attribute**

4252      A batch server that executes a batch job shall create, in the environment of the session leader of  
 4253      the batch job, each environment variable listed in the *Variable\_List* attribute of the batch job, and  
 4254      set the value of each such environment variable to that of the corresponding variable in the  
 4255      variable list.

4256    **3.2.2.2 Batch Job Routing**

4257      To route a batch job is to select a queue from a list and move the batch job to that queue.

4258      A batch server that has routing queues, which have been started, shall route the jobs in the  
 4259      routing queues owned by the batch server. A batch server may delay the routing of a batch job.  
 4260      The algorithm for selecting a batch job and the queue to which it will be routed is  
 4261      implementation-defined.

4262      When a routing queue has multiple possible destinations specified, then the precedence of the  
 4263      destinations is implementation-defined.

4264      A batch server that routes a batch job to a queue at another server shall move the batch job into  
 4265      the target queue with a *Queue Batch Job Request*.

4266      If the target server rejects the *Queue Batch Job Request*, the routing server shall retry routing the  
 4267      batch job or abort the batch job. A batch server that retries failed routings shall provide a means  
 4268      for the batch administrator to specify the number of retries and the minimum period of time  
 4269      between retries. The means by which an administrator specifies the number of retries and the  
 4270      delay between retries is implementation-defined. When the number of retries specified by the  
 4271      batch administrator has been exhausted, the batch server shall abort the batch job and perform  
 4272      the functions of *Batch Job Exit*; see Section 3.2.2.3.

4273    **3.2.2.3 Batch Job Exit**

4274      For each job in the EXITING state, the batch server that exited the batch job shall perform the  
 4275      following deferred services in the order specified:

- 4276      1. If buffering standard error, move that file into the location specified by the *Error\_Path*  
   4277      attribute of the batch job.
- 4278      2. If buffering standard output, move that file into the location specified by the *Output\_Path*  
   4279      attribute of the batch job.
- 4280      3. If the *Mail\_Points* attribute of the batch job includes MAIL\_AT\_EXIT, send mail to the users  
   4281      listed in the *Mail\_Users* attribute of the batch job. The mail message shall contain at least

4282                   the batch job identifier, queue, and server at which the batch job currently resides, and the  
4283                   *Job\_Owner* attribute.

4284                  4. Remove the batch job from the queue.

4285                   If a batch server that buffers the standard error output cannot return the standard error file to  
4286                   the standard error path at the time the batch job exits, the batch server shall do one of the  
4287                   following:

- 4288                   • Mail the standard error file to the batch job owner.
- 4289                   • Save the standard error file and mail the location and name of the file where the standard  
4290                   error is stored to the batch job owner.
- 4291                   • Save the standard error file and notify the user by other implementation-defined means.

4292                   If a batch server that buffers the standard output cannot return the standard output file to the  
4293                   standard output path at the time the batch job exits, the batch server shall do one of the  
4294                   following:

- 4295                   • Mail the standard output file to the batch job owner.
- 4296                   • Save the standard output file and mail the location and name of the file where the standard  
4297                   output is stored to the batch job owner.
- 4298                   • Save the standard output file and notify the user by other implementation-defined means.

4299                   At the conclusion of job exit processing, the batch job is no longer managed by a batch server.

4300                 3.2.2.4 *Batch Server Restart*

4301                   A batch server that has been either shutdown or terminated abnormally, and has returned to  
4302                   operation, is said to have “restarted”.

4303                   Upon restarting, a batch server shall requeue those jobs managed by the batch server that were  
4304                   in the RUNNING state at the time the batch server shut down and for which the *Rerunable*  
4305                   attribute of the batch job has the value TRUE.

4306                   Queues are defined to be non-volatile. A batch server shall store the content of queues that it  
4307                   controls in such a way that server and system shutdowns do not erase the content of the queues.

4308                 3.2.2.5 *Batch Job Abort*

4309                   A batch server that cannot perform a deferred service for a batch job shall abort the batch job.

4310                   A batch server that aborts a batch job shall perform the following services:

- 4311                   • Delete the batch job from the queue in which it resides.
- 4312                   • If the *Mail\_Points* attribute of the batch job includes the value MAIL\_AT\_ABORT, send mail  
4313                   to the users listed in the value of the *Mail\_Users* attribute of the job. The mail message shall  
4314                   contain at least the batch job identifier, queue, and server at which the batch job currently  
4315                   resides, the *Job\_Owner* attribute, and the reason for the abort.
- 4316                   • If the batch job was in the RUNNING state, terminate the session leader of the executing job  
4317                   by sending the session leader a SIGKILL, place the batch job in the EXITING state, and  
4318                   perform the actions of *Batch Job Exit*.

4319 **3.2.3 Requested Batch Services**

4320 This section describes the services provided by batch servers in response to requests from  
 4321 clients. Table 3-5 summarizes the current set of batch service requests and for each gives its type  
 4322 (deferred or not) and whether it is an optional function.

4323 **Table 3-5 Batch Services Summary**

| 4324 <b>Batch Service</b>         | 4325 <b>Deferred</b> | 4326 <b>Optional</b> |
|-----------------------------------|----------------------|----------------------|
| <i>Batch Job Execution</i>        | Yes                  | No                   |
| <i>Batch Job Routing</i>          | Yes                  | No                   |
| <i>Batch Job Exit</i>             | Yes                  | No                   |
| <i>Batch Server Restart</i>       | Yes                  | No                   |
| <i>Batch Job Abort</i>            | Yes                  | No                   |
| <i>Delete Batch Job Request</i>   | No                   | No                   |
| <i>Hold Batch Job Request</i>     | No                   | No                   |
| <i>Batch Job Message Request</i>  | No                   | Yes                  |
| <i>Batch Job Status Request</i>   | No                   | No                   |
| <i>Locate Batch Job Request</i>   | No                   | Yes                  |
| <i>Modify Batch Job Request</i>   | No                   | No                   |
| <i>Move Batch Job Request</i>     | No                   | No                   |
| <i>Queue Batch Job Request</i>    | No                   | No                   |
| <i>Batch Queue Status Request</i> | No                   | No                   |
| <i>Release Batch Job Request</i>  | No                   | No                   |
| <i>Rerun Batch Job Request</i>    | No                   | No                   |
| <i>Select Batch Jobs Request</i>  | No                   | No                   |
| <i>Server Shutdown Request</i>    | No                   | No                   |
| <i>Server Status Request</i>      | No                   | No                   |
| <i>Signal Batch Job Request</i>   | No                   | No                   |
| <i>Track Batch Job Request</i>    | No                   | Yes                  |

4346 If a request is rejected because the batch client is not authorized to perform the action, the batch  
 4347 server shall return the same status as when the batch job does not exist.

4348 **3.2.3.1 Delete Batch Job Request**

4349 A batch job is defined to have been deleted when it has been removed from the queue in which it  
 4350 resides and not instantiated in another queue. A client requests that the server that manages a  
 4351 batch job delete the batch job. Such a request is called a *Delete Batch Job Request*.

4352 A batch server shall reject a *Delete Batch Job Request* if any of the following statements are true:

- 4353 • The user of the batch client is not authorized to delete the designated job.
- 4354 • The designated job is not managed by the batch server.
- 4355 • The designated job is in a state inconsistent with the delete request.

4356 A batch server may reject a *Delete Batch Job Request* for other implementation-defined reasons.  
 4357 The method used to determine whether the user of a client is authorized to perform the  
 4358 requested action is implementation-defined.

4359 A batch server requested to delete a batch job shall delete the batch job if the batch job exists and  
 4360 is not in the EXITING state.

4361 A batch server that deletes a batch job in the RUNNING state shall send a SIGKILL signal to the  
 4362 session leader of the batch job. It is implementation-defined whether additional signals are sent

4363 to the session leader of the job prior to sending the SIGKILL signal.  
4364 A batch server that deletes a batch job in the RUNNING state shall place the batch job in the  
4365 EXITING state after it has killed the session leader of the batch job and shall perform the actions  
4366 of *Batch Job Exit*.

4367 3.2.3.2 *Hold Batch Job Request*

4368 A batch client can request that the batch server add one or more holds to a batch job. Such a  
4369 request is called a *Hold Batch Job Request*.

4370 A batch server shall reject a *Hold Batch Job Request* if any of the following statements are true:

- 4371 • The batch server does not support one or more of the requested holds to be added to the  
4372 batch job.
- 4373 • The user of the batch client is not authorized to add one or more of the requested holds to the  
4374 batch job.
- 4375 • The batch server does not manage the specified job.
- 4376 • The designated job is in the EXITING state.

4377 A batch server may reject a *Hold Batch Job Request* for other implementation-defined reasons. The  
4378 method used to determine whether the user of a client is authorized to perform the requested  
4379 action is implementation-defined.

4380 A batch server that accepts a *Hold Batch Job Request* for a batch job in the RUNNING state shall  
4381 place a hold on the batch job. The effects, if any, the hold will have on a batch job in the  
4382 RUNNING state are implementation-defined.

4383 A batch server that accepts a *Hold Batch Job Request* shall add each type of hold listed in the *Hold*  
4384 *Batch Job Request*, that is not already present, to the value of the *Hold\_Types* attribute of the batch  
4385 job.

4386 3.2.3.3 *Batch Job Message Request*

4387 *Batch Job Message Request* is an optional feature of batch servers. If an implementation supports  
4388 *Batch Job Message Request*, the statements in this section apply and the configuration variable  
4389 POSIX2\_PBS\_MESSAGE shall be set to 1.

4390 A batch client can request that a batch server write a message into certain output files of a batch  
4391 job. Such a request is called a *Batch Job Message Request*.

4392 A batch server shall reject a *Batch Job Message Request* if any of the following statements are true:

- 4393 • The batch server does not support sending messages to jobs.
- 4394 • The user of the batch client is not authorized to post a message to the designated job.
- 4395 • The designated job does not exist on the batch server.
- 4396 • The designated job is not in the RUNNING state.

4397 A batch server may reject a *Batch Job Message Request* for other implementation-defined reasons.  
4398 The method used to determine whether the user of a client is authorized to perform the  
4399 requested action is implementation-defined.

4400 A batch server that accepts a *Batch Job Message Request* shall write the message sent by the batch  
4401 client into the files indicated by the batch client.

4402 3.2.3.4 *Batch Job Status Request*

4403 A batch client can request that a batch server respond with the status and attributes of a batch  
4404 job. Such a request is called a *Batch Job Status Request*.

4405 A batch server shall reject a *Batch Job Status Request* if any of the following statements are true:

- 4406 • The user of the batch client is not authorized to query the status of the designated job.  
4407 • The designated job is not managed by the batch server.

4408 A batch server may reject a *Batch Job Status Request* for other implementation-defined reasons.  
4409 The method used to determine whether the user of a client is authorized to perform the  
4410 requested action is implementation-defined.

4411 A batch server that accepts a *Batch Job Status Request* shall return a *Batch Job Status Message* to the  
4412 batch client.

4413 A batch server may return other information in response to a *Batch Job Status Request*.

4414 3.2.3.5 *Locate Batch Job Request*

4415 *Locate Batch Job Request* is an optional feature of batch servers. If an implementation supports  
4416 *Locate Batch Job Request*, the statements in this section apply and the configuration variable  
4417 POSIX2\_PBS\_LOCATE shall be set to 1.

4418 A batch client can ask a batch server to respond with the location of a batch job that was created  
4419 by the batch server. Such a request is called a *Locate Batch Job Request*.

4420 A batch server that accepts a *Locate Batch Job Request* shall return a *Batch Job Location Message* to  
4421 the batch client.

4422 A batch server may reject a *Locate Batch Job Request* for a batch job that was not created by that  
4423 server.

4424 A batch server may reject a *Locate Batch Job Request* for a batch job that is no longer managed by  
4425 that server; that is, for a batch job that is not in a queue owned by that server.

4426 A batch server may reject a *Locate Batch Job Request* for other implementation-defined reasons.

4427 3.2.3.6 *Modify Batch Job Request*

4428 Batch clients modify (alter) the attributes of a batch job by making a request to the server that  
4429 manages the batch job. Such a request is called a *Modify Batch Job Request*.

4430 A batch server shall reject a *Modify Batch Job Request* if any of the following statements are true:

- 4431 • The user of the batch client is not authorized to make the requested modification to the batch  
4432 job.  
4433 • The designated job is not managed by the batch server.  
4434 • The requested modification is inconsistent with the state of the batch job.  
4435 • An unrecognized resource is requested for a batch job in an execution queue.

4436 A batch server may reject a *Modify Batch Job Request* for other implementation-defined reasons.  
4437 The method used to determine whether the user of a client is authorized to perform the  
4438 requested action is implementation-defined.

4439 A batch server that accepts a *Modify Batch Job Request* shall modify all the specified attributes of  
4440 the batch job. A batch server that rejects a *Modify Batch Job Request* shall modify none of the  
4441 attributes of the batch job.

4442 If the servicing by a batch server of an otherwise valid request would result in no change, then  
4443 the batch server shall indicate successful completion of the request.

4444 3.2.3.7 *Move Batch Job Request*

4445 A batch client can request that a batch server move a batch job to another destination. Such a  
4446 request is called a *Move Batch Job Request*.

4447 A batch server shall reject a *Move Batch Job Request* if any of the following statements are true:

- 4448 • The user of the batch client is not authorized to remove the designated job from the queue in  
4449 which the batch job resides.
- 4450 • The user of the batch client is not authorized to move the designated job to the destination.
- 4451 • The designated job is not managed by the batch server.
- 4452 • The designated job is in the EXITING state.
- 4453 • The destination is inaccessible.

4454 A batch server can reject a *Move Batch Job Request* for other implementation-defined reasons. The  
4455 method used to determine whether the user of a client is authorized to perform the requested  
4456 action is implementation-defined.

4457 A batch server that accepts a *Move Batch Job Request* shall perform the following services:

- 4458 • Queue the designated job at the destination.
- 4459 • Remove the designated job from the queue in which the batch job resides.

4460 If the destination resides on another batch server, the batch server shall queue the batch job at  
4461 the destination by sending a *Queue Batch Job Request* to the other server. If the *Queue Batch Job*  
4462 *Request* fails, the batch server shall reject the *Move Batch Job Request*. If the *Queue Batch Job Request*  
4463 succeeds, the batch server shall remove the batch job from its queue.

4464 The batch server shall not modify any attributes of the batch job.

4465 3.2.3.8 *Queue Batch Job Request*

4466 A batch queue is controlled by one and only one batch server. A batch server is said to own the  
4467 queues that it controls. Batch clients make requests of batch servers to have jobs queued. Such a  
4468 request is called a *Queue Batch Job Request*.

4469 A batch server requested to queue a batch job for which the queue is not specified shall select an  
4470 implementation-defined queue for the batch job. Such a queue is called the “default queue” of  
4471 the batch server. The implementation shall provide the means for a batch administrator to  
4472 specify the default queue. The queue, whether specified or defaulted, is called the “target  
4473 queue”.

4474 A batch server shall reject a *Queue Batch Job Request* if any of the following statements are true:

- 4475 • The client is not authorized to create a batch job in the target queue.
- 4476 • The request specifies a queue that does not exist on the batch server.
- 4477 • The target queue is an execution queue and the batch server cannot satisfy a resource  
4478 requirement of the batch job.
- 4479 • The target queue is an execution queue and an unrecognized resource is requested.
- 4480 • The target queue is an execution queue, the batch server does not support checkpointing, and  
4481 the value of the *Checkpoint* attribute of the batch job is not NO\_CHECKPOINT.

- 4482     • The job requires access to a user identifier that the batch client is not authorized to access.

4483 A batch server may reject a *Queue Batch Job Request* for other implementation-defined reasons.

4484 A batch server that accepts a *Queue Batch Job Request* for a batch job for which the  
4485 PBS\_O\_QUEUE value is missing from the value of the *Variable\_List* attribute of the batch job  
4486 shall add that variable to the list and set the value to the name of the target queue. Once set, no  
4487 server shall change the value of PBS\_O\_QUEUE, even if the batch job is moved to another  
4488 queue.

4489 A batch server that accepts a *Queue Batch Job Request* for a batch job for which the PBS\_JOBID  
4490 value is missing from the value of the *Variable\_List* attribute shall add that variable to the list and  
4491 set the value to the batch job identifier assigned by the server in the format:

4492       sequence\_number.server

4493 A batch server that accepts a *Queue Batch Job Request* for a batch job for which the PBS\_JOBNAME value is missing from the value of the *Variable\_List* attribute of the batch job shall add that variable to the list and set the value to the *Job\_Name* attribute of the batch job.

4496 3.2.3.9 *Batch Queue Status Request*

4497 A batch client can request that a batch server respond with the status and attributes of a queue.  
4498 Such a request is called a *Batch Queue Status Request*.

4499 A batch server shall reject a *Batch Queue Status Request* if any of the following statements are true:

- 4500     • The user of the batch client is not authorized to query the status of the designated queue.  
4501     • The designated queue does not exist on the batch server.

4502 A batch server may reject a *Batch Queue Status Request* for other implementation-defined reasons.  
4503 The method used to determine whether the user of a client is authorized to perform the  
4504 requested action is implementation-defined.

4505 A batch server that accepts a *Batch Queue Status Request* shall return a *Batch Queue Status Reply* to  
4506 the batch client.

4507 3.2.3.10 *Release Batch Job Request*

4508 A batch client can request that the server remove one or more holds from a batch job. Such a  
4509 request is called a *Release Batch Job Request*.

4510 A batch server shall reject a *Release Batch Job Request* if any of the following statements are true:

- 4511     • The user of the batch client is not authorized to remove one or more of the requested holds  
4512       from the batch job.  
4513     • The batch server does not manage the specified job.

4514 A batch server may reject a *Release Batch Job Request* for other implementation-defined reasons.  
4515 The method used to determine whether the user of a client is authorized to perform the  
4516 requested action is implementation-defined.

4517 A batch server that accepts a *Release Batch Job Request* shall remove each type of hold listed in the  
4518 *Release Batch Job Request*, that is present, from the value of the *Hold\_Types* attribute of the batch  
4519 job.

4520 3.2.3.11 *Rerun Batch Job Request*

4521 To rerun a batch job is to kill the session leader of the batch job and leave the batch job eligible  
4522 for re-execution. A batch client can request that a batch server rerun a batch job. Such a request is  
4523 called *Rerun Batch Job Request*.

4524 A batch server shall reject a *Rerun Batch Job Request* if any of the following statements are true:

- 4525 • The user of the batch client is not authorized to rerun the designated job.
- 4526 • The *Rerunable* attribute of the designated job has the value FALSE.
- 4527 • The designated job is not in the RUNNING state.
- 4528 • The batch server does not manage the designated job.

4529 A batch server may reject a *Rerun Batch Job Request* for other implementation-defined reasons.  
4530 The method used to determine whether the user of a client is authorized to perform the  
4531 requested action is implementation-defined.

4532 A batch server that rejects a *Rerun Batch Job Request* shall in no way modify the execution of the  
4533 batch job.

4534 A batch server that accepts a request to rerun a batch job shall perform the following services:

- 4535 • Requeue the batch job in the execution queue in which it was executing.
- 4536 • Send a SIGKILL signal to the process group of the session leader of the batch job.

4537 An implementation may indicate to the batch job owner that the batch job has been rerun.  
4538 Whether and how the batch job owner is notified that a batch job is rerun is implementation-  
4539 defined.

4540 A batch server that reruns a batch job may send other implementation-defined signals to the  
4541 session leader of the batch job prior to sending the SIGKILL signal.

4542 A batch server may preferentially select a rerun job for execution. Whether rerun jobs shall be  
4543 selected for execution before other jobs is implementation-defined.

4544 3.2.3.12 *Select Batch Jobs Request*

4545 A batch client can request from a batch server a list of jobs managed by that server that match a  
4546 list of selection criteria. Such a request is called a *Select Batch Jobs Request*. All the batch jobs  
4547 managed by the batch server that receives the request are candidates for selection.

4548 A batch server that accepts a *Select Batch Jobs Request* shall return a list of zero or more job  
4549 identifiers that correspond to jobs that meet the selection criteria.

4550 If the batch client is not authorized to query the status of a batch job, the batch server shall not  
4551 select the batch job.

4552 3.2.3.13 *Server Shutdown Request*

4553 A batch server is defined to have shut down when it does not respond to requests from clients  
4554 and does not perform deferred services for jobs. A batch client can request that a batch server  
4555 shut down. Such a request is called a *Server Shutdown Request*.

4556 A batch server shall reject a *Server Shutdown Request* from a client that is not authorized to shut  
4557 down the batch server. The method used to determine whether the user of a client is authorized  
4558 to perform the requested action is implementation-defined.

4559 A batch server may reject a *Server Shutdown Request* for other implementation-defined reasons.  
4560 The reasons for which a *Server Shutdown Request* may be rejected are implementation-defined.

4561 At server shutdown, a batch server shall do, in order of preference, one of the following:

- 4562 • If checkpointing is implemented and the batch job is checkpointable, then checkpoint the  
4563 batch job and requeue it.
- 4564 • If the batch job is rerunnable, then requeue the batch job to be rerun (restarted from the  
4565 beginning).
- 4566 • Abort the batch job.

4567 **3.2.3.14 Server Status Request**

4568 A batch client can request that a batch server respond with the status and attributes of the batch  
4569 server. Such a request is called a *Server Status Request*.

4570 A batch server shall reject a *Server Status Request* if the following statement is true:

- 4571 • The user of the batch client is not authorized to query the status of the designated server.

4572 A batch server may reject a *Server Status Request* for other implementation-defined reasons. The  
4573 method used to determine whether the user of a client is authorized to perform the requested  
4574 action is implementation-defined.

4575 A batch server that accepts a *Server Status Request* shall return a *Server Status Reply* to the batch  
4576 client.

4577 **3.2.3.15 Signal Batch Job Request**

4578 A batch client can request that a batch server signal the session leader of a batch job. Such a  
4579 request is called a *Signal Batch Job Request*.

4580 A batch server shall reject a *Signal Batch Job Request* if any of the following statements are true:

- 4581 • The user of the batch client is not authorized to signal the batch job.
- 4582 • The job is not in the RUNNING state.
- 4583 • The batch server does not manage the designated job.
- 4584 • The requested signal is not supported by the implementation.

4585 A batch server may reject a *Signal Batch Job Request* for other implementation-defined reasons.  
4586 The method used to determine whether the user of a client is authorized to perform the  
4587 requested action is implementation-defined.

4588 A batch server that accepts a request to signal a batch job shall send the signal requested by the  
4589 batch client to the process group of the session leader of the batch job.

4590 **3.2.3.16 Track Batch Job Request**

4591 *Track Batch Job Request* is an optional feature of batch servers. If an implementation supports  
4592 *Track Batch Job Request*, the statements in this section apply and the configuration variable  
4593 `POSIX2_PBS_TRACK` shall be set to 1.

4594 *Track Batch Job Request* provides a method for tracking the current location of a batch job. Clients  
4595 may use the tracking information to determine the batch server that should receive a batch  
4596 server request.

4597 If *Track Batch Job Request* is supported by a batch server, then when the batch server queues a  
4598 batch job as a result of a *Queue Batch Job Request*, and the batch server is not the batch server that  
4599 created the batch job, the batch server shall send a *Track Batch Job Request* to the batch server that  
4600 created the job.

4601 If *Track Batch Job Request* is supported by a batch server, then the *Track Batch Job Request* may also  
4602 be sent to other servers as a backup to the primary server. The method by which backup servers  
4603 are specified is implementation-defined.

4604 If *Track Batch Job Request* is supported by a batch server that receives a *Track Batch Job Request*,  
4605 then the batch server shall record the current location of the batch job as contained in the  
4606 request.

### 4607 3.3 Common Behavior for Batch Environment Utilities

#### 4608 3.3.1 Batch Job Identifier

4609 A utility shall recognize *job\_identifiers* of the format:

4610 [sequence\_number] [.server\_name] [@server]

4611 where:

4612 *sequence\_number* An integer that, when combined with *server\_name*, provides a batch job  
4613 identifier that is unique within the batch system.

4614 *server\_name* The name of the batch server to which the batch job was originally submitted.

4615 *server* The name of the batch server that is currently managing the batch job.

4616 If the application omits the batch *server\_name* portion of a batch job identifier, a utility shall use  
4617 the name of a default batch server.

4618 If the application omits the batch *server* portion of a batch job identifier, a utility shall use:

- 4619 • The batch server indicated by *server\_name*, if present
- 4620 • The name of the default batch server
- 4621 • The name of the batch server that is currently managing the batch job

4622 If only *@server* is specified, then the status of all jobs owned by the user on the requested server  
4623 is listed.

4624 The means by which a utility determines the default batch server is implementation-defined.

4625 If the application presents the batch *server* portion of a batch job identifier to a utility, the utility  
4626 shall send the request to the specified server.

4627 A strictly conforming application shall use the syntax described for the job identifier. Whenever  
4628 a batch job identifier is specified whose syntax is not recognized by an implementation, then a  
4629 message for each error that occurs shall be written to standard error and the utility shall exit  
4630 with an exit status greater than zero.

4631 When a batch job identifier is supplied as an argument to a batch utility and the *server\_name*  
4632 portion of the batch job identifier is omitted, then the utility shall use the name of the default  
4633 batch server.

4634 When a batch job identifier is supplied as an argument to a batch utility and the batch *server*  
4635 portion of the batch job identifier is omitted, then the utility shall use either:

- 4636           • The name of the default batch server  
 4637           or:  
 4638           • The name of the batch server that is currently managing the batch job  
 4639 When a batch job identifier is supplied as an argument to a batch utility and the batch *server*  
 4640 portion of the batch job identifier is specified, then the utility shall send the required *Batch Server*  
 4641 *Request* to the specified server.

### 4642 3.3.2 Destination

4643 The utility shall recognize a *destination* of the format:

4644        [*queue*] [*@server*]

4645 where:

- 4646        *queue*           The name of a valid execution or routing queue at the batch server denoted by  
 4647                        *@server*, defined as a string of up to 15 alphanumeric characters in the portable  
 4648                        character set (see the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 4649                        6.1, Portable Character Set) where the first character is alphabetic.  
 4650        *server*          The name of a batch server, defined as a string of alphanumeric characters in  
 4651                        the portable character set.

4652 If the application omits the batch *server* portion of a destination, then the utility shall use either:

- 4653           • The name of the default batch server  
 4654           or:  
 4655           • The name of the batch server that is currently managing the batch job

4656 The means by which a utility determines the default batch server is implementation-defined.

4657 If the application omits the *queue* portion of a destination, then the utility shall use the name of  
 4658                        the default queue at the batch server chosen. The means by which a batch server determines its  
 4659                        default queue is implementation-defined. If a destination is specified in the *queue@server* form,  
 4660                        then the utility shall use the specified queue at the specified server.

4661 A strictly conforming application shall use the syntax described for a destination. Whenever a  
 4662                        destination is specified whose syntax is not recognized by an implementation, then a message  
 4663                        shall be written to standard error and the utility shall exit with an exit status greater than zero.

### 4664 3.3.3 Multiple Keyword-Value Pairs

4665 For each option that can have multiple keyword-value pair arguments, the following rules shall  
 4666 apply. Examples of options that can have list-oriented option-arguments are **-u value@keyword**  
 4667 and **-I keyword=value**.

- 4668 1. If a batch utility is presented with a list-oriented option-argument for which a keyword has  
 4669                        a corresponding value that begins with a single or double quote, then the utility shall stop  
 4670                        interpreting the input stream for delimiters until a second single or double quote,  
 4671                        respectively, is encountered. This feature allows some flexibility for a comma (',') or  
 4672                        equals sign ('=') to be part of the value string for a particular keyword; for example:

4673        keywd1='val1,val2',keywd2="val3,val4"

4674 **Note:** This may require the user to escape the quotes as in the following command:

- 4675                                 foo -xkeywd1=\`val1,val2\`,keywd2=\`val3,val4\`
- 4676     2. If a batch server is presented with a list-oriented attribute that has a keyword that was  
4677        encountered earlier in the list, then the later entry for that keyword shall replace the earlier  
4678        entry.
- 4679     3. If a batch server is presented with a list-oriented attribute that has a keyword without any  
4680        corresponding value of the form *keyword*= or @*keyword* and the same keyword was  
4681        encountered earlier in the list, then the prior entry for that keyword shall be ignored by the  
4682        batch server.
- 4683     4. If a batch utility is expecting a list-oriented option-argument entry of the form  
4684        *keyword*=*value*, but is presented with an entry of the form *keyword* without any  
4685        corresponding *value*, then the entry shall be treated as though a default value of NULL was  
4686        assigned (that is, *keyword*=NULL) for entry parsing purposes. The utility shall include only  
4687        the keyword, not the NULL value, in the associated job attribute.
- 4688     5. If a batch utility is expecting a list-oriented option-argument entry of the form  
4689        *value*@*keyword*, but is presented with an entry of the form *value* without any corresponding  
4690        *keyword*, then the entry shall be treated as though a keyword of NULL was assigned (that  
4691        is, *value*@NULL) for entry parsing purposes. The utility shall include only the value, not  
4692        the NULL keyword, in the associated job attribute.
- 4693     6. A batch server shall accept a list-oriented attribute that has multiple occurrences of the  
4694        same keyword, interpreting the keywords, in order, with the last value encountered taking  
4695        precedence over prior instances of the same keyword. This rule allows, but does not  
4696        require, a batch utility to preprocess the attribute to remove duplicate keywords.
- 4697     7. If a batch utility is presented with multiple list-oriented option-arguments on the  
4698        command line or in script directives, or both, for a single option, then the utility shall  
4699        concatenate, in order, any command line keyword and value pairs to the end of any  
4700        directive keyword and value pairs separated by a single comma to produce a single string  
4701        that is an equivalent, valid option-argument. The resulting string shall be assigned to the  
4702        associated attribute of the batch job (after optionally removing duplicate entries as  
4703        described in item 6).

## **Utilities**

- 4705 This chapter contains the definitions of the utilities, as follows:
- 4706 • Mandatory utilities that are present on every conformant system
- 4707 • Optional utilities that are present only on systems supporting the associated option; see  
4708      Section 1.8.1 (on page 9) for information on the options in this volume of  
4709      IEEE Std 1003.1-2001

## 4710 NAME

4711 admin — create and administer SCCS files (**DEVELOPMENT**)

## 4712 SYNOPSIS

```
4713 XSI admin -i[name][-n][-a login][-d flag][-e login][-f flag][-m mrlist]
4714 [-r rel][-t[name][-y[comment]]] newfile
4715
4716 admin -n[-a login][-d flag][-e login][-f flag][-m mrlist][-t[name]]
4717 [-y[comment]] newfile ...
4718
4719 admin [-a login][-d flag][-m mrlist][-r rel][-t[name]] file ...
4720
4721 admin -h file ...
4722
4723 admin -z file ...
```

## 4721 DESCRIPTION

4722 The *admin* utility shall create new SCCS files or change parameters of existing ones. If a named  
4723 file does not exist, it shall be created, and its parameters shall be initialized according to the  
4724 specified options. Parameters not initialized by an option shall be assigned a default value. If a  
4725 named file does exist, parameters corresponding to specified options shall be changed, and other  
4726 parameters shall be left as is.

4727 All SCCS filenames supplied by the application shall be of the form *s.filename*. New SCCS files  
4728 shall be given read-only permission mode. Write permission in the parent directory is required  
4729 to create a file. All writing done by *admin* shall be to a temporary *x-file*, named *x.filename* (see *get*)  
4730 created with read-only mode if *admin* is creating a new SCCS file, or created with the same mode  
4731 as that of the SCCS file if the file already exists. After successful execution of *admin*, the SCCS file  
4732 shall be removed (if it exists), and the *x-file* shall be renamed with the name of the SCCS file. This  
4733 ensures that changes are made to the SCCS file only if no errors occur.

4734 The *admin* utility shall also use a transient lock file (named *z.filename*), which is used to prevent  
4735 simultaneous updates to the SCCS file; see *get*.

## 4736 OPTIONS

4737 The *admin* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
4738 12.2, Utility Syntax Guidelines, except that the **-i**, **-t**, and **-y** options have optional option-  
4739 arguments. These optional option-arguments shall not be presented as separate arguments. The  
4740 following options are supported:

- 4741   **-n**       Create a new SCCS file. When **-n** is used without **-i**, the SCCS file shall be created  
4742       with control information but without any file data.
- 4743   **-i[name]**   Specify the *name* of a file from which the text for a new SCCS file shall be taken.  
4744       The text constitutes the first delta of the file (see the **-r** option for the delta  
4745       numbering scheme). If the **-i** option is used, but the *name* option-argument is  
4746       omitted, the text shall be obtained by reading the standard input. If this option is  
4747       omitted, the SCCS file shall be created with control information but without any  
4748       file data. The **-i** option implies the **-n** option.
- 4749   **-r SID**     Specify the SID of the initial delta to be inserted. This SID shall be a trunk SID; that  
4750       is, the branch and sequence numbers shall be zero or missing. The level number is  
4751       optional, and defaults to 1.
- 4752   **-t[name]**   Specify the *name* of a file from which descriptive text for the SCCS file shall be  
4753       taken. In the case of existing SCCS files (neither **-i** nor **-n** is specified):

- 4754 • A **-t** option without a *name* option-argument shall cause the removal of  
 4755 descriptive text (if any) currently in the SCCS file.
- 4756 • A **-t** option with a *name* option-argument shall cause the text (if any) in the  
 4757 named file to replace the descriptive text (if any) currently in the SCCS file.
- 4758       **-f flag**      Specify a *flag*, and, possibly, a value for the *flag*, to be placed in the SCCS file.  
 4759       Several **-f** options may be supplied on a single *admin* command line.  
 4760       Implementations shall recognize the following flags and associated values:
- 4761       **b**       Allow use of the **-b** option on a *get* command to create branch deltas.
- 4762       **cceil**   Specify the highest release (that is, ceiling), a number less than or equal to  
 4763       9 999, which may be retrieved by a *get* command for editing. The default  
 4764       value for an unspecified **c** flag shall be 9 999.
- 4765       **ffloor**   Specify the lowest release (that is, floor), a number greater than 0 but less  
 4766       than 9 999, which may be retrieved by a *get* command for editing. The  
 4767       default value for an unspecified **f** flag shall be 1.
- 4768       **dSID**   Specify the default delta number (SID) to be used by a *get* command.
- 4769       **istr**     Treat the “No ID keywords” message issued by *get* or *delta* as a fatal  
 4770       error. In the absence of this flag, the message is only a warning. The  
 4771       message is issued if no SCCS identification keywords (see *get*) are found  
 4772       in the text retrieved or stored in the SCCS file. If a value is supplied, the  
 4773       application shall ensure that the keywords exactly match the given string;  
 4774       however, the string shall contain a keyword, and no embedded  
 4775       <newline>s.
- 4776       **j**       Allow concurrent *get* commands for editing on the same SID of an SCCS  
 4777       file. This allows multiple concurrent updates to the same version of the  
 4778       SCCS file.
- 4779       **llist**   Specify a *list* of releases to which deltas can no longer be made (that is, *get*  
 4780       **-e** against one of these locked releases fails). Conforming applications  
 4781       shall use the following syntax to specify a *list*. Implementations may  
 4782       accept additional forms as an extension:
- ```
4783 <list> ::= a | <range-list>
4784 <range-list> ::= <range> | <range-list>, <range>
4785 <range> ::= <SID>
```
- 4786 The character *a* in the *list* shall be equivalent to specifying all releases for
 4787 the named SCCS file. The non-terminal *<SID>* in *range* shall be the delta
 4788 number of an existing delta associated with the SCCS file.
- 4789 **n** Cause *delta* to create a null delta in each of those releases (if any) being
 4790 skipped when a delta is made in a new release (for example, in making
 4791 delta 5.1 after delta 2.7, releases 3 and 4 are skipped). These null deltas
 4792 shall serve as anchor points so that branch deltas may later be created
 4793 from them. The absence of this flag shall cause skipped releases to be
 4794 nonexistent in the SCCS file, preventing branch deltas from being created
 4795 from them in the future. During the initial creation of an SCCS file, the **n**
 4796 flag may be ignored; that is, if the **-r** option is used to set the release
 4797 number of the initial SID to a value greater than 1, null deltas need not be
 4798 created for the “skipped” releases.

4799	qtext	Substitute user-definable <i>text</i> for all occurrences of the %Q% keyword in the SCCS file text retrieved by <i>get</i> .
4800		
4801	mmod	Specify the module name of the SCCS file substituted for all occurrences of the %M% keyword in the SCCS file text retrieved by <i>get</i> . If the m flag is not specified, the value assigned shall be the name of the SCCS file with the leading '.' removed.
4802		
4803		
4804		
4805	ttype	Specify the <i>type</i> of module in the SCCS file substituted for all occurrences of the %Y% keyword in the SCCS file text retrieved by <i>get</i> .
4806		
4807	vpgm	Cause <i>delta</i> to prompt for modification request (MR) numbers as the reason for creating a delta. The optional value specifies the name of an MR number validation program. (If this flag is set when creating an SCCS file, the application shall ensure that the m option is also used even if its value is null.)
4808		
4809		
4810		
4811		
4812	-d flag	Remove (delete) the specified <i>flag</i> from an SCCS file. Several -d options may be supplied on a single <i>admin</i> command. See the -f option for allowable <i>flag</i> names. (The llist flag gives a <i>list</i> of releases to be unlocked. See the -f option for further description of the l flag and the syntax of a <i>list</i> .)
4813		
4814		
4815		
4816	-a login	Specify a <i>login</i> name, or numerical group ID, to be added to the list of users who may make deltas (changes) to the SCCS file. A group ID shall be equivalent to specifying all <i>login</i> names common to that group ID. Several -a options may be used on a single <i>admin</i> command line. As many <i>logins</i> , or numerical group IDs, as desired may be on the list simultaneously. If the list of users is empty, then anyone may add deltas. If <i>login</i> or group ID is preceded by a '!', the users so specified shall be denied permission to make deltas.
4817		
4818		
4819		
4820		
4821		
4822		
4823	-e login	Specify a <i>login</i> name, or numerical group ID, to be erased from the list of users allowed to make deltas (changes) to the SCCS file. Specifying a group ID is equivalent to specifying all <i>login</i> names common to that group ID. Several -e options may be used on a single <i>admin</i> command line.
4824		
4825		
4826		
4827	-y[comment]	Insert the <i>comment</i> text into the SCCS file as a comment for the initial delta in a manner identical to that of <i>delta</i> . In the POSIX locale, omission of the -y option shall result in a default comment line being inserted in the form:
4828		
4829		
4830		"date and time created %s %s by %s", <date>, <time>, <login>
4831		where <date> is expressed in the format of the <i>date</i> utility's %y/%m/%d conversion specification, <time> in the format of the <i>date</i> utility's %T conversion specification format, and <login> is the login name of the user creating the file.
4832		
4833		
4834	-m mrlist	Insert the list of modification request (MR) numbers into the SCCS file as the reason for creating the initial delta in a manner identical to <i>delta</i> . The application shall ensure that the v flag is set and the MR numbers are validated if the v flag has a value (the name of an MR number validation program). A diagnostic message shall be written if the v flag is not set or MR validation fails.
4835		
4836		
4837		
4838		
4839	-h	Check the structure of the SCCS file and compare the newly computed checksum with the checksum that is stored in the SCCS file. If the newly computed checksum does not match the checksum in the SCCS file, a diagnostic message shall be written.
4840		
4841		
4842		
4843	-z	Recompute the SCCS file checksum and store it in the first line of the SCCS file (see the -h option above). Note that use of this option on a truly corrupted file may
4844		

4845 prevent future detection of the corruption.

4846 OPERANDS

4847 The following operands shall be supported:

4848 *file* A pathname of an existing SCCS file or a directory. If *file* is a directory, the *admin* utility shall behave as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the pathname does not begin with **s.**) and unreadable files shall be silently ignored.

4852 *newfile* A pathname of an SCCS file to be created.

4853 If exactly one *file* or *newfile* operand appears, and it is '**-**', the standard input shall be read; each line of the standard input shall be taken to be the name of an SCCS file to be processed. Non-SCCS files and unreadable files shall be silently ignored.

4856 STDIN

4857 The standard input shall be a text file used only if **-i** is specified without an option-argument or if a *file* or *newfile* operand is specified as '**-**'. If the first character of any standard input line is <SOH> in the POSIX locale, the results are unspecified.

4860 INPUT FILES

4861 The existing SCCS files shall be text files of an unspecified format.

4862 The application shall ensure that the file named by the **-i** option's *name* option-argument shall be a text file; if the first character of any line in this file is <SOH> in the POSIX locale, the results are unspecified. If this file contains more than 99 999 lines, the number of lines recorded in the header for this file shall be 99 999 for this delta.

4866 ENVIRONMENT VARIABLES

4867 The following environment variables shall affect the execution of *admin*:

4868 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

4872 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

4874 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

4877 *LC_MESSAGES*

4878 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and the contents of the default **-y** comment.

4881 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

4882 ASYNCHRONOUS EVENTS

4883 Default.

4884 STDOUT

4885 Not used.

4886 STDERR

4887 The standard error shall be used only for diagnostic messages.

4888 OUTPUT FILES

4889 Any SCCS files created shall be text files of an unspecified format. During processing of a *file*, a
4890 locking *z-file*, as described in *get* (on page 473), may be created and deleted.

4891 EXTENDED DESCRIPTION

4892 None.

4893 EXIT STATUS

4894 The following exit values shall be returned:

4895 0 Successful completion.

4896 >0 An error occurred.

4897 CONSEQUENCES OF ERRORS

4898 Default.

4899 APPLICATION USAGE

4900 It is recommended that directories containing SCCS files be writable by the owner only, and that
4901 SCCS files themselves be read-only. The mode of the directories should allow only the owner to
4902 modify SCCS files contained in the directories. The mode of the SCCS files prevents any
4903 modification at all except by SCCS commands.

4904 EXAMPLES

4905 None.

4906 RATIONALE

4907 None.

4908 FUTURE DIRECTIONS

4909 None.

4910 SEE ALSO

4911 *delta, get, prs, what*

4912 CHANGE HISTORY

4913 First released in Issue 2.

4914 Issue 6

4915 The normative text is reworded to avoid use of the term “must” for application requirements.

4916 The normative text is reworded to emphasize the term “shall” for implementation requirements.

4917 The grammar is updated.

4918 The Open Group Base Resolution bwg2001-007 is applied, adding new text to the INPUT FILES
4919 section warning that the maximum lines recorded in the file is 99 999.

4920 The Open Group Base Resolution bwg2001-009 is applied, amending the description of the **-h**
4921 option.

4922 NAME

4923 alias — define or display aliases

4924 SYNOPSIS

4925 UP alias [*alias-name[=string]* . . .]

4926

4927 DESCRIPTION

4928 The *alias* utility shall create or redefine alias definitions or write the values of existing alias
4929 definitions to standard output. An alias definition provides a string value that shall replace a
4930 command name when it is encountered; see Section 2.3.1 (on page 32).

4931 An alias definition shall affect the current shell execution environment and the execution
4932 environments of the subshells of the current shell. When used as specified by this volume of
4933 IEEE Std 1003.1-2001, the alias definition shall not affect the parent process of the current shell
4934 nor any utility environment invoked by the shell; see Section 2.12 (on page 61).

4935 OPTIONS

4936 None.

4937 OPERANDS

4938 The following operands shall be supported:

4939 *alias-name* Write the alias definition to standard output.

4940 *alias-name*=*string*

4941 Assign the value of *string* to the alias *alias-name*.

4942 If no operands are given, all alias definitions shall be written to standard output.

4943 STDIN

4944 Not used.

4945 INPUT FILES

4946 None.

4947 ENVIRONMENT VARIABLES

4948 The following environment variables shall affect the execution of *alias*:

4949 *LANG* Provide a default value for the internationalization variables that are unset or null.
4950 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
4951 Internationalization Variables for the precedence of internationalization variables
4952 used to determine the values of locale categories.)

4953 *LC_ALL* If set to a non-empty string value, override the values of all the other
4954 internationalization variables.

4955 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
4956 characters (for example, single-byte as opposed to multi-byte characters in
4957 arguments).

4958 *LC_MESSAGES*

4959 Determine the locale that should be used to affect the format and contents of
4960 diagnostic messages written to standard error.

4961 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

4962 **ASYNCHRONOUS EVENTS**

4963 Default.

4964 **STDOUT**4965 The format for displaying aliases (when no operands or only *name* operands are specified) shall
4966 be:4967 "%s=%s\n", *name*, *value*4968 The *value* string shall be written with appropriate quoting so that it is suitable for reinput to the
4969 shell. See the description of shell quoting in Section 2.2 (on page 30).4970 **STDERR**

4971 The standard error shall be used only for diagnostic messages.

4972 **OUTPUT FILES**

4973 None.

4974 **EXTENDED DESCRIPTION**

4975 None.

4976 **EXIT STATUS**

4977 The following exit values shall be returned:

4978 0 Successful completion.

4979 >0 One of the *name* operands specified did not have an alias definition, or an error occurred.4980 **CONSEQUENCES OF ERRORS**

4981 Default.

4982 **APPLICATION USAGE**

4983 None.

4984 **EXAMPLES**

- 4985 1. Change
- ls*
- to give a columnated, more annotated output:

4986 alias ls="ls -CF"

- 4987 2. Create a simple “redo” command to repeat previous entries in the command history file:

4988 alias r='fc -s'

- 4989 3. Use 1K units for
- du*
- :

4990 alias du=du\ -k

- 4991 4. Set up
- nohup*
- so that it can deal with an argument that is itself an alias name:

4992 alias nohup="nohup "

4993 **RATIONALE**4994 The *alias* description is based on historical KornShell implementations. Known differences exist
4995 between that and the C shell. The KornShell version was adopted to be consistent with all the
4996 other KornShell features in this volume of IEEE Std 1003.1-2001, such as command line editing.4997 Since *alias* affects the current shell execution environment, it is generally provided as a shell
4998 regular built-in.4999 Historical versions of the KornShell have allowed aliases to be exported to scripts that are
5000 invoked by the same shell. This is triggered by the *alias -x* flag; it is allowed by this volume of
5001 IEEE Std 1003.1-2001 only when an explicit extension such as *-x* is used. The standard
5002 developers considered that aliases were of use primarily to interactive users and that they

5003 should normally not affect shell scripts called by those users; functions are available to such
5004 scripts.

5005 Historical versions of the KornShell had not written aliases in a quoted manner suitable for
5006 reentry to the shell, but this volume of IEEE Std 1003.1-2001 has made this a requirement for all
5007 similar output. Therefore, consistency with this volume of IEEE Std 1003.1-2001 was chosen over
5008 this detail of historical practice.

5009 **FUTURE DIRECTIONS**

5010 None.

5011 **SEE ALSO**

5012 Section 2.9.5 (on page 54)

5013 **CHANGE HISTORY**

5014 First released in Issue 4.

5015 **Issue 6**

5016 This utility is marked as part of the User Portability Utilities option.

5017 The APPLICATION USAGE section is added.

5018 **NAME**

5019 ar — create and maintain library archives

5020 **SYNOPSIS**

5021 SD ar -d[-v] archive file ...

5022

5023 XSI ar -m[-abiv][posname] archive file ...

5024

5025 XSI ar -p[-v][-s]archive [file ...]

5026 XSI ar -q[-cv] archive file ...

5027

5028 XSI ar -r[-cuv][-abi][posname]archive file ...

5029 XSI ar -t[-v][-s]archive [file ...]

5030 XSI ar -x[-v][-sCT]archive [file ...]

5031 **DESCRIPTION**

5032 The *ar* utility is part of the Software Development Utilities option.

5033 The *ar* utility can be used to create and maintain groups of files combined into an archive. Once
5034 an archive has been created, new files can be added, and existing files in an archive can be
5035 extracted, deleted, or replaced. When an archive consists entirely of valid object files, the
5036 implementation shall format the archive so that it is usable as a library for link editing (see *c99*
5037 and *fort77*). When some of the archived files are not valid object files, the suitability of the
5038 XSI archive for library use is undefined. If an archive consists entirely of printable files, the entire
5039 archive shall be printable.

5040 When *ar* creates an archive, it creates administrative information indicating whether a symbol
5041 table is present in the archive. When there is at least one object file that *ar* recognizes as such in
5042 the archive, an archive symbol table shall be created in the archive and maintained by *ar*; it is
5043 used by the link editor to search the archive. Whenever the *ar* utility is used to create or update
5044 the contents of such an archive, the symbol table shall be rebuilt. The *-s* option shall force the
5045 symbol table to be rebuilt.

5046 All *file* operands can be pathnames. However, files within archives shall be named by a filename,
5047 which is the last component of the pathname used when the file was entered into the archive.
5048 The comparison of *file* operands to the names of files in archives shall be performed by
5049 comparing the last component of the operand to the name of the file in the archive.

5050 It is unspecified whether multiple files in the archive may be identically named. In the case of
5051 XSI such files, however, each *file* and *posname* operand shall match only the first file in the archive
5052 having a name that is the same as the last component of the operand.

5053 **OPTIONS**

5054 The *ar* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
5055 Utility Syntax Guidelines.

5056 The following options shall be supported:

5057 XSI **-a** Position new files in the archive after the file named by the *posname* operand.

5058 XSI **-b** Position new files in the archive before the file named by the *posname* operand.

5059 **-c** Suppress the diagnostic message that is written to standard error by default when
5060 the archive *archive* is created.

5061 XSI	-C	Prevent extracted files from replacing like-named files in the file system. This option is useful when -T is also used, to prevent truncated filenames from replacing files with the same prefix.
5064	-d	Delete one or more <i>files</i> from <i>archive</i> .
5065 XSI	-i	Position new files in the archive before the file in the archive named by the <i>posname</i> operand (equivalent to -b).
5067 XSI	-m	Move the named files in the archive. The -a , -b , or -i options with the <i>posname</i> operand indicate the position; otherwise, move the names files in the archive to the end of the archive.
5070	-p	Write the contents of the <i>files</i> in the archive named by <i>file</i> operands from <i>archive</i> to the standard output. If no <i>file</i> operands are specified, the contents of all files in the archive shall be written in the order of the archive.
5073 XSI	-q	Append the named files to the end of the archive. In this case <i>ar</i> does not check whether the added files are already in the archive. This is useful to bypass the searching otherwise done when creating a large archive piece by piece.
5076	-r	Replace or add <i>files</i> to <i>archive</i> . If the archive named by <i>archive</i> does not exist, a new archive shall be created and a diagnostic message shall be written to standard error (unless the -c option is specified). If no <i>files</i> are specified and the <i>archive</i> exists, the results are undefined. Files that replace existing files in the archive shall not change the order of the archive. Files that do not replace existing files in the archive shall be appended to the archive unless a -a , -b , or -i option specifies another position.
5081 XSI	-s	Force the regeneration of the archive symbol table even if <i>ar</i> is not invoked with an option that modifies the archive contents. This option is useful to restore the archive symbol table after it has been stripped; see <i>strip</i> .
5086	-t	Write a table of contents of <i>archive</i> to the standard output. The files specified by the <i>file</i> operands shall be included in the written list. If no <i>file</i> operands are specified, all files in <i>archive</i> shall be included in the order of the archive.
5089 XSI	-T	Allow filename truncation of extracted files whose archive names are longer than the file system can support. By default, extracting a file with a name that is too long shall be an error; a diagnostic message shall be written and the file shall not be extracted.
5093	-u	Update older files in the archive. When used with the -r option, files in the archive shall be replaced only if the corresponding <i>file</i> has a modification time that is at least as new as the modification time of the file in the archive.
5096	-v	Give verbose output. When used with the option characters -d , -r , or -x , write a detailed file-by-file description of the archive creation and maintenance activity, as described in the STDOUT section.
5099		When used with -p , write the name of the file in the archive to the standard output before writing the file in the archive itself to the standard output, as described in the STDOUT section.
5102		When used with -t , include a long listing of information about the files in the archive, as described in the STDOUT section.
5104	-x	Extract the files in the archive named by the <i>file</i> operands from <i>archive</i> . The contents of the archive shall not be changed. If no <i>file</i> operands are given, all files

5106 in the archive shall be extracted. The modification time of each file extracted shall
5107 be set to the time the file is extracted from the archive.

5108 OPERANDS

5109 The following operands shall be supported:

5110 *archive* A pathname of the archive.

5111 *file* A pathname. Only the last component shall be used when comparing against the
5112 names of files in the archive. If two or more *file* operands have the same last
5113 pathname component (basename), the results are unspecified. The
5114 implementation's archive format shall not truncate valid filenames of files added
5115 to or replaced in the archive.

5116 XSI *posname* The name of a file in the archive, used for relative positioning; see options **-m** and
5117 **-r**.

5118 STDIN

5119 Not used.

5120 INPUT FILES

5121 The archive named by *archive* shall be a file in the format created by *ar -r*.

5122 ENVIRONMENT VARIABLES

5123 The following environment variables shall affect the execution of *ar*:

5124 *LANG* Provide a default value for the internationalization variables that are unset or null.
5125 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
5126 Internationalization Variables for the precedence of internationalization variables
5127 used to determine the values of locale categories.)

5128 *LC_ALL* If set to a non-empty string value, override the values of all the other
5129 internationalization variables.

5130 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
5131 characters (for example, single-byte as opposed to multi-byte characters in
5132 arguments and input files).

5133 *LC_MESSAGES*

5134 Determine the locale that should be used to affect the format and contents of
5135 diagnostic messages written to standard error.

5136 *LC_TIME* Determine the format and content for date and time strings written by *ar -tv*.

5137 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

5138 *TMPDIR* Determine the pathname that overrides the default directory for temporary files, if
5139 any.

5140 *TZ* Determine the timezone used to calculate date and time strings written by *ar -tv*.
5141 If *TZ* is unset or null, an unspecified default timezone shall be used.

5142 ASYNCHRONOUS EVENTS

5143 Default.

5144 STDOUT

5145 If the **-d** option is used with the **-v** option, the standard output format shall be:

5146 "*d* - %s\n", <*file*>

5147 where *file* is the operand specified on the command line.

5148 If the **-p** option is used with the **-v** option, *ar* shall precede the contents of each file with:
5149 "*\n<%s>\n\n*", <*file*>
5150 where *file* is the operand specified on the command line, if *file* operands were specified, and the
5151 name of the file in the archive if they were not.
5152 If the **-r** option is used with the **-v** option:
5153 • If *file* is already in the archive, the standard output format shall be:
5154 "r - %s\n", <*file*>
5155 where <*file*> is the operand specified on the command line.
5156 • If *file* is not already in the archive, the standard output format shall be:
5157 "a - %s\n", <*file*>
5158 where <*file*> is the operand specified on the command line.
5159 If the **-t** option is used, *ar* shall write the names of the files in the archive to the standard output
5160 in the format:
5161 "%s\n", <*file*>
5162 where *file* is the operand specified on the command line, if *file* operands were specified, or the
5163 name of the file in the archive if they were not.
5164 If the **-t** option is used with the **-v** option, the standard output format shall be:
5165 "%s %u/%u %u %s %d %d:%d %d %s\n", <*member mode*>, <*user ID*>,
5166 <*group ID*>, <*number of bytes in member*>,
5167 <*abbreviated month*>, <*day-of-month*>, <*hour*>,
5168 <*minute*>, <*year*>, <*file*>
5169 where:
5170 <*file*> Shall be the operand specified on the command line, if *file* operands were specified,
5171 or the name of the file in the archive if they were not.
5172 <*member mode*> Shall be formatted the same as the <*file mode*> string defined in the STDOUT
5173 section of *ls*, except that the first character, the <*entry type*>, is not used; the string
5174 represents the file mode of the file in the archive at the time it was added to or
5175 replaced in the archive.
5176 The following represent the last-modification time of a file when it was most recently added to
5177 or replaced in the archive:
5178 <*abbreviated month*> Equivalent to the format of the %b conversion specification format in *date*.
5179 <*day-of-month*> Equivalent to the format of the %e conversion specification format in *date*.
5180 <*hour*> Equivalent to the format of the %H conversion specification format in *date*.
5181 <*minute*> Equivalent to the format of the %M conversion specification format in *date*.
5182 <*year*> Equivalent to the format of the %Y conversion specification format in *date*.
5183 When *LC_TIME* does not specify the POSIX locale, a different format and order of presentation
5184 of these fields relative to each other may be used in a format appropriate in the specified locale.

5188 If the **-x** option is used with the **-v** option, the standard output format shall be:

5189 "x - %s\n", <file>

5190 where *file* is the operand specified on the command line, if *file* operands were specified, or the
5191 name of the file in the archive if they were not.

5192 **STDERR**

5193 The standard error shall be used only for diagnostic messages. The diagnostic message about
5194 creating a new archive when **-c** is not specified shall not modify the exit status.

5195 **OUTPUT FILES**

5196 Archives are files with unspecified formats.

5197 **EXTENDED DESCRIPTION**

5198 None.

5199 **EXIT STATUS**

5200 The following exit values shall be returned:

5201 0 Successful completion.

5202 >0 An error occurred.

5203 **CONSEQUENCES OF ERRORS**

5204 Default.

5205 **APPLICATION USAGE**

5206 None.

5207 **EXAMPLES**

5208 None.

5209 **RATIONALE**

5210 The archive format is not described. It is recognized that there are several known ar formats,
5211 which are not compatible. The ar utility is included, however, to allow creation of archives that
5212 are intended for use only on one machine. The archive is specified as a file, and it can be moved
5213 as a file. This does allow an archive to be moved from one machine to another machine that uses
5214 the same implementation of ar.

5215 Utilities such as pax (and its forebears tar and cpio) also provide portable “archives”. This is a not
5216 a duplication; the ar utility is included to provide an interface primarily for make and the
5217 compilers, based on a historical model.

5218 In historical implementations, the **-q** option (available on XSI-conforming systems) is known to
5219 execute quickly because ar does not check on whether the added members are already in the
5220 archive. This is useful to bypass the searching otherwise done when creating a large archive
5221 piece-by-piece. These remarks may but need not remain true for a brand new implementation of
5222 this utility; hence, these remarks have been moved into the RATIONALE.

5223 BSD implementations historically required applications to provide the **-s** option whenever the
5224 archive was supposed to contain a symbol table. As in this volume of IEEE Std 1003.1-2001,
5225 System V historically creates or updates an archive symbol table whenever an object file is
5226 removed from, added to, or updated in the archive.

5227 The OPERANDS section requires what might seem to be true without specifying it: the archive
5228 cannot truncate the filenames below {NAME_MAX}. Some historical implementations do so,
5229 however, causing unexpected results for the application. Therefore, this volume of
5230 IEEE Std 1003.1-2001 makes the requirement explicit to avoid misunderstandings.

5231 According to the System V documentation, the options **-dmpqrtx** are not required to begin with
5232 a hyphen ('-'). This volume of IEEE Std 1003.1-2001 requires that a conforming application use
5233 the leading hyphen.

5234 The archive format used by the 4.4 BSD implementation is documented in this RATIONALE as
5235 an example:

5236 A file created by *ar* begins with the "magic" string "**!<arch>\n**". The rest of the archive is
5237 made up of objects, each of which is composed of a header for a file, a possible filename, and
5238 the file contents. The header is portable between machine architectures, and, if the file
5239 contents are printable, the archive is itself printable.

5240 The header is made up of six ASCII fields, followed by a two-character trailer. The fields are
5241 the object name (16 characters), the file last modification time (12 characters), the user and
5242 group IDs (each 6 characters), the file mode (8 characters), and the file size (10 characters). All
5243 numeric fields are in decimal, except for the file mode, which is in octal.

5244 The modification time is the file *st_mtime* field. The user and group IDs are the file *st_uid* and
5245 *st_gid* fields. The file mode is the file *st_mode* field. The file size is the file *st_size* field. The
5246 two-byte trailer is the string "**<newline>**".

5247 Only the name field has any provision for overflow. If any filename is more than 16
5248 characters in length or contains an embedded space, the string "#1/" followed by the ASCII
5249 length of the name is written in the name field. The file size (stored in the archive header) is
5250 incremented by the length of the name. The name is then written immediately following the
5251 archive header.

5252 Any unused characters in any of these fields are written as **<space>**s. If any fields are their
5253 particular maximum number of characters in length, there is no separation between the
5254 fields.

5255 Objects in the archive are always an even number of bytes long; files that are an odd number
5256 of bytes long are padded with a **<newline>**, although the size in the header does not reflect
5257 this.

5258 The *ar* utility description requires that (when all its members are valid object files) *ar* produce an
5259 object code library, which the linkage editor can use to extract object modules. If the linkage
5260 editor needs a symbol table to permit random access to the archive, *ar* must provide it; however,
5261 *ar* does not require a symbol table.

5262 The BSD **-o** option was omitted. It is a rare conforming application that uses *ar* to extract object
5263 code from a library with concern for its modification time, since this can only be of importance
5264 to *make*. Hence, since this functionality is not deemed important for applications portability, the
5265 modification time of the extracted files is set to the current time.

5266 There is at least one known implementation (for a small computer) that can accommodate only
5267 object files for that system, disallowing mixed object and other files. The ability to handle any
5268 type of file is not only historical practice for most implementations, but is also a reasonable
5269 expectation.

5270 Consideration was given to changing the output format of *ar -tv* to the same format as the
5271 output of *ls -l*. This would have made parsing the output of *ar* the same as that of *ls*. This was
5272 rejected in part because the current *ar* format is commonly used and changes would break
5273 historical usage. Second, *ar* gives the user ID and group ID in numeric format separated by a
5274 slash. Changing this to be the user name and group name would not be correct if the archive
5275 were moved to a machine that contained a different user database. Since *ar* cannot know
5276 whether the archive was generated on the same machine, it cannot tell what to report.

5277 The text on the **-ur** option combination is historical practice—since one filename can easily
5278 represent two different files (for example, **/a/foo** and **/b/foo**), it is reasonable to replace the file in
5279 the archive even when the modification time in the archive is identical to that in the file system.

5280 **FUTURE DIRECTIONS**

5281 None.

5282 **SEE ALSO**

5283 *c99, date, fort77, pax, strip* the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 13,
5284 Headers, <unistd.h> description of {POSIX_NO_TRUNC}

5285 **CHANGE HISTORY**

5286 First released in Issue 2.

5287 **Issue 5**

5288 The FUTURE DIRECTIONS section is added.

5289 **Issue 6**

5290 This utility is marked as part of the Software Development Utilities option.

5291 The STDOUT description is changed for the **-v** option to align with the IEEE P1003.2b draft
5292 standard.

5293 The normative text is reworded to avoid use of the term “must” for application requirements.

5294 The **TZ** entry is added to the ENVIRONMENT VARIABLES section.

5295 IEEE PASC Interpretation 1003.2 #198 is applied, changing the description to consistently use
5296 “file” to refer to a file in the file system hierarchy, “archive” to refer to the archive being
5297 operated upon by the **ar** utility, and “file in the archive” to refer to a copy of a file that is
5298 contained in the archive.

5299 NAME

5300 asa — interpret carriage-control characters

5301 SYNOPSIS

5302 FR asa [*file* ...]

5304 DESCRIPTION

5305 The *asa* utility shall write its input files to standard output, mapping carriage-control characters
5306 from the text files to line-printer control sequences in an implementation-defined manner.

5307 The first character of every line shall be removed from the input, and the following actions are
5308 performed.

5309 If the character removed is:

5310 <space> The rest of the line is output without change.

5311 0 A <newline> is output, then the rest of the input line.

5312 1 One or more implementation-defined characters that causes an advance to the next
5313 page shall be output, followed by the rest of the input line.

5314 + The <newline> of the previous line shall be replaced with one or more
5315 implementation-defined characters that causes printing to return to column position 1,
5316 followed by the rest of the input line. If the '+' is the first character in the input, it shall
5317 be equivalent to <space>.

5318 The action of the *asa* utility is unspecified upon encountering any character other than those
5319 listed above as the first character in a line.

5320 OPTIONS

5321 None.

5322 OPERANDS

5323 *file* A pathname of a text file used for input. If no *file* operands are specified, the
5324 standard input shall be used.

5325 STDIN

5326 The standard input shall be used only if no *file* operands are specified; see the INPUT FILES
5327 section.

5328 INPUT FILES

5329 The input files shall be text files.

5330 ENVIRONMENT VARIABLES

5331 The following environment variables shall affect the execution of *asa*:

5332 *LANG* Provide a default value for the internationalization variables that are unset or null.
5333 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
5334 Internationalization Variables for the precedence of internationalization variables
5335 used to determine the values of locale categories.)

5336 *LC_ALL* If set to a non-empty string value, override the values of all the other
5337 internationalization variables.

5338 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
5339 characters (for example, single-byte as opposed to multi-byte characters in
5340 arguments and input files).

5341 ***LC_MESSAGES***
5342 Determine the locale that should be used to affect the format and contents of
5343 diagnostic messages written to standard error.

5344 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

5345 **ASYNCHRONOUS EVENTS**
5346 Default.

5347 **STDOUT**
5348 The standard output shall be the text from the input file modified as described in the
5349 DESCRIPTION section.

5350 **STDERR**
5351 None.

5352 **OUTPUT FILES**
5353 None.

5354 **EXTENDED DESCRIPTION**
5355 None.

5356 **EXIT STATUS**
5357 The following exit values shall be returned:
5358 0 All input files were output successfully.
5359 >0 An error occurred.

5360 **CONSEQUENCES OF ERRORS**
5361 Default.

5362 **APPLICATION USAGE**
5363 None.

5364 **EXAMPLES**
5365 1. The following command:
5366 `asa file`
5367 permits the viewing of *file* (created by a program using FORTRAN-style carriage-control
5368 characters) on a terminal.
5369 2. The following command:
5370 `a.out | asa | lp`
5371 formats the FORTRAN output of **a.out** and directs it to the printer.

5372 **RATIONALE**
5373 The *asa* utility is needed to map “standard” FORTRAN 77 output into a form acceptable to
5374 contemporary printers. Usually, *asa* is used to pipe data to the *lp* utility; see *lp*.
5375 This utility is generally used only by FORTRAN programs. The standard developers decided to
5376 retain *asa* to avoid breaking the historical large base of FORTRAN applications that put
5377 carriage-control characters in their output files. There is no requirement that a system have a
5378 FORTRAN compiler in order to run applications that need *asa*.
5379 Historical implementations have used an ASCII <form-feed> in response to a 1 and an ASCII
5380 <carriage-return> in response to a '+'. It is suggested that implementations treat characters
5381 other than 0, 1, and '+' as <space> in the absence of any compelling reason to do otherwise.
5382 However, the action is listed here as “unspecified”, permitting an implementation to provide

5383 extensions to access fast multiple-line slewing and channel seeking in a non-portable manner.

5384 **FUTURE DIRECTIONS**

5385 None.

5386 **SEE ALSO**

5387 *fort77, lp*

5388 **CHANGE HISTORY**

5389 First released in Issue 4.

5390 **Issue 6**

5391 This utility is marked as part of the FORTRAN Runtime Utilities option.

5392 The normative text is reworded to avoid use of the term “must” for application requirements.

5393 NAME

5394 at — execute commands at a later time

5395 SYNOPSIS

5396 UP at [-m][-f file][-q queuename] -t time_arg

5397 at [-m][-f file][-q queuename] timespec ...

5398 at -r at_job_id ...

5399 at -l -q queuename

5400 at -l [at_job_id ...]

5401

5402 DESCRIPTION

5403 The *at* utility shall read commands from standard input and group them together as an *at-job*, to
5404 be executed at a later time.

5405 The *at-job* shall be executed in a separate invocation of the shell, running in a separate process
5406 group with no controlling terminal, except that the environment variables, current working
5407 directory, file creation mask, and other implementation-defined execution-time attributes in
5408 effect when the *at* utility is executed shall be retained and used when the *at-job* is executed.

5409 When the *at-job* is submitted, the *at_job_id* and scheduled time shall be written to standard error.
5410 The *at_job_id* is an identifier that shall be a string consisting solely of alphanumeric characters
5411 and the period character. The *at_job_id* shall be assigned by the system when the job is scheduled
5412 such that it uniquely identifies a particular job.

5413 User notification and the processing of the job's standard output and standard error are
5414 described under the **-m** option.

5415 XSI Users shall be permitted to use *at* if their name appears in the file **/usr/lib/cron/at.allow**. If that
5416 file does not exist, the file **/usr/lib/cron/at.deny** shall be checked to determine whether the user
5417 shall be denied access to *at*. If neither file exists, only a process with the appropriate privileges
5418 shall be allowed to submit a job. If only **at.deny** exists and is empty, global usage shall be
5419 permitted. The **at.allow** and **at.deny** files shall consist of one user name per line.

5420 OPTIONS

5421 The *at* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
5422 Utility Syntax Guidelines.

5423 The following options shall be supported:

5424 **-f file** Specify the pathname of a file to be used as the source of the *at-job*, instead of
5425 standard input.

5426 **-l** (The letter ell.) Report all jobs scheduled for the invoking user if no *at_job_id*
5427 operands are specified. If *at_job_ids* are specified, report only information for these
5428 jobs. The output shall be written to standard output.

5429 **-m** Send mail to the invoking user after the *at-job* has run, announcing its completion.
5430 Standard output and standard error produced by the *at-job* shall be mailed to the
5431 user as well, unless redirected elsewhere. Mail shall be sent even if the job
5432 produces no output.

5433 If **-m** is not used, the job's standard output and standard error shall be provided to
5434 the user by means of mail, unless they are redirected elsewhere; if there is no such
5435 output to provide, the implementation need not notify the user of the job's
5436 completion.

- 5437 **-q queueName**
 5438 Specify in which queue to schedule a job for submission. When used with the **-l**
 5439 option, limit the search to that particular queue. By default, at-jobs shall be
 5440 scheduled in queue *a*. In contrast, queue *b* shall be reserved for batch jobs; see
 5441 *batch*. The meanings of all other *queueNames* are implementation-defined. If **-q** is
 5442 specified along with either of the **-t time_arg** or *timespec* arguments, the results are
 5443 unspecified.
- 5444 **-r**
 5445 Remove the jobs with the specified *at_job_id* operands that were previously
 5446 scheduled by the *at* utility.
- 5447 **-t time_arg** Submit the job to be run at the time specified by the *time* option-argument, which
 5447 the application shall ensure has the format as specified by the *touch -t time* utility.

5448 OPERANDS

5449 The following operands shall be supported:

- 5450 **at_job_id** The name reported by a previous invocation of the *at* utility at the time the job was
 5451 scheduled.
- 5452 **timespec** Submit the job to be run at the date and time specified. All of the *timespec* operands
 5453 are interpreted as if they were separated by <space>s and concatenated, and shall
 5454 be parsed as described in the grammar at the end of this section. The date and time
 5455 shall be interpreted as being in the timezone of the user (as determined by the *TZ*
 5456 variable), unless a timezone name appears as part of *time*, below.

5457 In the POSIX locale, the following describes the three parts of the time
 5458 specification string. All of the values from the *LC_TIME* categories in the POSIX
 5459 locale shall be recognized in a case-insensitive manner.

- 5460 **time** The time can be specified as one, two, or four digits. One-digit and
 5461 two-digit numbers shall be taken to be hours; four-digit numbers to
 5462 be hours and minutes. The time can alternatively be specified as two
 5463 numbers separated by a colon, meaning *hour:min*. An AM/PM
 5464 indication (one of the values from the **am_pm** keywords in the
 5465 *LC_TIME* locale category) can follow the time; otherwise, a 24-hour
 5466 clock time shall be understood. A timezone name can also follow to
 5467 further qualify the time. The acceptable timezone names are
 5468 implementation-defined, except that they shall be case-insensitive
 5469 and the string **utc** is supported to indicate the time is in Coordinated
 5470 Universal Time. In the POSIX locale, the *time* field can also be one of
 5471 the following tokens:

- 5472 **midnight** Indicates the time 12:00 am (00:00).
- 5473 **noon** Indicates the time 12:00 pm.
- 5474 **now** Indicates the current day and time. Invoking *at <now>*
 5475 shall submit an at-job for potentially immediate
 5476 execution (that is, subject only to unspecified
 5477 scheduling delays).

- 5478 **date** An optional *date* can be specified as either a month name (one of the
 5479 values from the **mon** or **abmon** keywords in the *LC_TIME* locale
 5480 category) followed by a day number (and possibly year number
 5481 preceded by a comma), or a day of the week (one of the values from
 5482 the **day** or **abday** keywords in the *LC_TIME* locale category). In the
 5483 POSIX locale, two special days shall be recognized:

5484 **today** Indicates the current day.

5485 **tomorrow** Indicates the day following the current day.

5486 If no *date* is given, **today** shall be assumed if the given time is greater
5487 than the current time, and **tomorrow** shall be assumed if it is less. If
5488 the given month is less than the current month (and no year is given),
5489 next year shall be assumed.

5490 *increment* The optional *increment* shall be a number preceded by a plus sign
5491 ('+') and suffixed by one of the following: **minutes**, **hours**, **days**,
5492 **weeks**, **months**, or **years**. (The singular forms shall also be
5493 accepted.) The keyword **next** shall be equivalent to an increment
5494 number of +1. For example, the following are equivalent commands:

5495 at 2pm + 1 week
5496 at 2pm next week

5497 The following grammar describes the precise format of *timespec* in the POSIX locale. The general
5498 conventions for this style of grammar are described in Section 1.10 (on page 19). This formal
5499 syntax shall take precedence over the preceding text syntax description. The longest possible
5500 token or delimiter shall be recognized at a given point. When used in a *timespec*, white space
5501 shall also delimit tokens.

5502 %token hr24clock_hr_min
5503 %token hr24clock_hour
5504 /*
5505 An hr24clock_hr_min is a one, two, or four-digit number. A one-digit
5506 or two-digit number constitutes an hr24clock_hour. An hr24clock_hour
5507 may be any of the single digits [0,9], or may be double digits, ranging
5508 from [00,23]. If an hr24clock_hr_min is a four-digit number, the
5509 first two digits shall be a valid hr24clock_hour, while the last two
5510 represent the number of minutes, from [00,59].
5511 */
5512 %token wallclock_hr_min
5513 %token wallclock_hour
5514 /*
5515 A wallclock_hr_min is a one, two-digit, or four-digit number.
5516 A one-digit or two-digit number constitutes a wallclock_hour.
5517 A wallclock_hour may be any of the single digits [1,9], or may
5518 be double digits, ranging from [01,12]. If a wallclock_hr_min
5519 is a four-digit number, the first two digits shall be a valid
5520 wallclock_hour, while the last two represent the number of
5521 minutes, from [00,59].
5522 */
5523 %token minute
5524 /*
5525 A minute is a one or two-digit number whose value can be [0,9]
5526 or [00,59].
5527 */
5528 %token day_number
5529 /*
5530 A day_number is a number in the range appropriate for the particular
5531 month and year specified by month_name and year_number, respectively.

```
5532     If no year_number is given, the current year is assumed if the given
5533     date and time are later this year. If no year_number is given and
5534     the date and time have already occurred this year and the month is
5535     not the current month, next year is the assumed year.
5536 */
5537 %token year_number
5538 /*
5539     A year_number is a four-digit number representing the year A.D., in
5540     which the at_job is to be run.
5541 */
5542 %token inc_number
5543 /*
5544     The inc_number is the number of times the succeeding increment
5545     period is to be added to the specified date and time.
5546 */
5547 %token timezone_name
5548 /*
5549     The name of an optional timezone suffix to the time field, in an
5550     implementation-defined format.
5551 */
5552 %token month_name
5553 /*
5554     One of the values from the mon or abmon keywords in the LC_TIME
5555     locale category.
5556 */
5557 %token day_of_week
5558 /*
5559     One of the values from the day or abday keywords in the LC_TIME
5560     locale category.
5561 */
5562 %token am_pm
5563 /*
5564     One of the values from the am_pm keyword in the LC_TIME locale
5565     category.
5566 */
5567 %start timespec
5568 %%
5569 timespec      : time
5570           | time date
5571           | time increment
5572           | time date increment
5573           | nowspec
5574           ;
5575 nowspec       : "now"
5576           | "now" increment
5577           ;
5578 time         : hr24clock_hr_min
5579           | hr24clock_hr_min timezone_name
```

```
5580 | hr24clock_hour ":" minute
5581 | hr24clock_hour ":" minute timezone_name
5582 | wallclock_hr_min am_pm
5583 | wallclock_hr_min am_pm timezone_name
5584 | wallclock_hour ":" minute am_pm
5585 | wallclock_hour ":" minute am_pm timezone_name
5586 | "noon"
5587 | "midnight"
5588 ;
5589 date : month_name day_number
5590 | month_name day_number ", " year_number
5591 | day_of_week
5592 | "today"
5593 | "tomorrow"
5594 ;
5595 increment : "+" inc_number inc_period
5596 | "next" inc_period
5597 ;
5598 inc_period : "minute" | "minutes"
5599 | "hour" | "hours"
5600 | "day" | "days"
5601 | "week" | "weeks"
5602 | "month" | "months"
5603 | "year" | "years"
5604 ;
```

5605 STDIN

5606 The standard input shall be a text file consisting of commands acceptable to the shell command
5607 language described in Chapter 2 (on page 29). The standard input shall only be used if no **-f file**
5608 option is specified.

5609 INPUT FILES

5610 See the STDIN section.

5611 XSI The text files **/usr/lib/cron/at.allow** and **/usr/lib/cron/at.deny** shall contain zero or more user
5612 names, one per line, of users who are, respectively, authorized or denied access to the **at** and
5613 **batch** utilities.

5614 ENVIRONMENT VARIABLES

5615 The following environment variables shall affect the execution of **at**:

5616 **LANG** Provide a default value for the internationalization variables that are unset or null.
5617 (See the **Base Definitions** volume of IEEE Std 1003.1-2001, Section 8.2,
5618 Internationalization Variables for the precedence of internationalization variables
5619 used to determine the values of locale categories.)

5620 **LC_ALL** If set to a non-empty string value, override the values of all the other
5621 internationalization variables.

5622 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
5623 characters (for example, single-byte as opposed to multi-byte characters in
5624 arguments and input files).

5625 **LC_MESSAGES**

5626 Determine the locale that should be used to affect the format and contents of

5627		diagnostic messages written to standard error and informative messages written to standard output.
5629	XSI	NLSPATH Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
5630		LC_TIME Determine the format and contents for date and time strings written and accepted by <i>at</i> .
5632		SHELL Determine a name of a command interpreter to be used to invoke the <i>at</i> -job. If the variable is unset or null, <i>sh</i> shall be used. If it is set to a value other than a name for <i>sh</i> , the implementation shall do one of the following: use that shell; use <i>sh</i> ; use the login shell from the user database; or any of the preceding accompanied by a warning diagnostic about which was chosen.
5637		TZ Determine the timezone. The job shall be submitted for execution at the time specified by <i>timespec</i> or -t <i>time</i> relative to the timezone specified by the TZ variable. If <i>timespec</i> specifies a timezone, it shall override TZ . If <i>timespec</i> does not specify a timezone and TZ is unset or null, an unspecified default timezone shall be used.
5642		ASYNCHRONOUS EVENTS
5643		Default.
5644		STDOUT
5645		When standard input is a terminal, prompts of unspecified format for each line of the user input described in the STDIN section may be written to standard output.
5647		In the POSIX locale, the following shall be written to the standard output for each job when jobs are listed in response to the -l option:
5649		"%s\t%s\n", <i>at_job_id</i> , < <i>date</i> >
5650		where <i>date</i> shall be equivalent in format to the output of:
5651		<i>date</i> +"%a %b %e %T %Y"
5652		The date and time written shall be adjusted so that they appear in the timezone of the user (as determined by the TZ variable).
5654		STDERR
5655		In the POSIX locale, the following shall be written to standard error when a job has been successfully submitted:
5657		" job %s at %s\n", <i>at_job_id</i> , < <i>date</i> >
5658		where <i>date</i> has the same format as that described in the STDOUT section. Neither this, nor warning messages concerning the selection of the command interpreter, shall be considered a diagnostic that changes the exit status.
5661		Diagnostic messages, if any, shall be written to standard error.
5662		OUTPUT FILES
5663		None.
5664		EXTENDED DESCRIPTION
5665		None.
5666		EXIT STATUS
5667		The following exit values shall be returned:
5668		0 The <i>at</i> utility successfully submitted, removed, or listed a job or jobs.

5669 >0 An error occurred.

5670 CONSEQUENCES OF ERRORS

5671 The job shall not be scheduled, removed, or listed.

5672 APPLICATION USAGE

5673 The format of the at command line shown here is guaranteed only for the POSIX locale. Other
5674 cultures may be supported with substantially different interfaces, although implementations are
5675 encouraged to provide comparable levels of functionality.

5676 Since the commands run in a separate shell invocation, running in a separate process group with
5677 no controlling terminal, open file descriptors, traps, and priority inherited from the invoking
5678 environment are lost.

5679 Some implementations do not allow substitution of different shells using *SHELL*. System V
5680 systems, for example, have used the login shell value for the user in /etc/passwd. To select
5681 reliably another command interpreter, the user must include it as part of the script, such as:

```
5682 $ at 1800
5683 myshell myscript
5684 EOT
5685 job ... at ...
5686 $
```

5687 EXAMPLES

1. This sequence can be used at a terminal:

```
5689 at -m 0730 tomorrow
5690 sort < file >outfile
5691 EOT
```

2. This sequence, which demonstrates redirecting standard error to a pipe, is useful in a command procedure (the sequence of output redirection specifications is significant):

```
5694 at now + 1 hour <<!
5695 diff file1 file2 2>&1 >outfile | mailx mygroup
5696 !
```

3. To have a job reschedule itself, at can be invoked from within the at-job. For example, this daily processing script named **my.daily** runs every day (although crontab is a more appropriate vehicle for such work):

```
5700 # my.daily runs every day
5701 daily processing
5702 at now tomorrow < my.daily
```

4. The spacing of the three portions of the POSIX locale *timespec* is quite flexible as long as there are no ambiguities. Examples of various times and operand presentation include:

```
5705 at 0815am Jan 24
5706 at 8 :15amjan24
5707 at now "+ 1day"
5708 at 5 pm FRIday
5709 at '17
5710     utc+
5711     30minutes'
```

RATIONALE

5712 The *at* utility reads from standard input the commands to be executed at a later time. It may be
5713 useful to redirect standard output and standard error within the specified commands.

5715 The **-t time** option was added as a new capability to support an internationalized way of
5716 specifying a time for execution of the submitted job.

5717 Early proposals added a “jobname” concept as a way of giving submitted jobs names that are
5718 meaningful to the user submitting them. The historical, system-specified *at_job_id* gives no
5719 indication of what the job is. Upon further reflection, it was decided that the benefit of this was
5720 not worth the change in historical interface. The *at* functionality is useful in simple
5721 environments, but in large or complex situations, the functionality provided by the Batch
5722 Services option is more suitable.

5723 The **-q** option historically has been an undocumented option, used mainly by the *batch* utility.

5724 The System V **-m** option was added to provide a method for informing users that an at-job had
5725 completed. Otherwise, users are only informed when output to standard error or standard
5726 output are not redirected.

5727 The behavior of *at <now>* was changed in an early proposal from being unspecified to
5728 submitting a job for potentially immediate execution. Historical BSD *at* implementations
5729 support this. Historical System V implementations give an error in that case, but a change to the
5730 System V versions should have no backwards-compatibility ramifications.

5731 On BSD-based systems, a **-u user** option has allowed those with appropriate privileges to access
5732 the work of other users. Since this is primarily a system administration feature and is not
5733 universally implemented, it has been omitted. Similarly, a specification for the output format for
5734 a user with appropriate privileges viewing the queues of other users has been omitted.

5735 The **-f file** option from System V is used instead of the BSD method of using the last operand as
5736 the pathname. The BSD method is ambiguous—does:

5737 *at 1200 friday*

5738 mean the same thing if there is a file named **friday** in the current directory?

5739 The *at_job_id* is composed of a limited character set in historical practice, and it is mandated here
5740 to invalidate systems that might try using characters that require shell quoting or that could not
5741 be easily parsed by shell scripts.

5742 The *at* utility varies between System V and BSD systems in the way timezones are used. On
5743 System V systems, the *TZ* variable affects the at-job submission times and the times displayed
5744 for the user. On BSD systems, *TZ* is not taken into account. The BSD behavior is easily achieved
5745 with the current specification. If the user wishes to have the timezone default to that of the
5746 system, they merely need to issue the *at* command immediately following an unsetting or null
5747 assignment to *TZ*. For example:

5748 *TZ= at noon ...*

5749 gives the desired BSD result.

5750 While the yacc-like grammar specified in the OPERANDS section is lexically unambiguous with
5751 respect to the digit strings, a lexical analyzer would probably be written to look for and return
5752 digit strings in those cases. The parser could then check whether the digit string returned is a
5753 valid *day_number*, *year_number*, and so on, based on the context.

5754 **FUTURE DIRECTIONS**

5755 None.

5756 **SEE ALSO**5757 *batch, crontab*5758 **CHANGE HISTORY**

5759 First released in Issue 2.

5760 **Issue 6**

5761 This utility is marked as part of the User Portability Utilities option.

5762 The following new requirements on POSIX implementations derive from alignment with the
5763 Single UNIX Specification:

- 5764 • If **-m** is not used, the job's standard output and standard error are provided to the user by
5765 mail.

5766 The effects of using the **-q** and **-t** options as defined in the IEEE P1003.2b draft standard are
5767 specified.

5768 The normative text is reworded to avoid use of the term "must" for application requirements.

5769 NAME

5770 awk — pattern scanning and processing language

5771 **SYNOPSIS**

5772 awk [-F ERE] [-v assignment] ... program [argument ...]

5773 awk [-F ERE] -f progfile ... [-v assignment] ... [argument ...]

5774 DESCRIPTION

5775 The *awk* utility shall execute programs written in the *awk* programming language, which is
5776 specialized for textual data manipulation. An *awk* program is a sequence of patterns and
5777 corresponding actions. When input is read that matches a pattern, the action associated with
5778 that pattern is carried out.

5779 Input shall be interpreted as a sequence of records. By default, a record is a line, less its
5780 terminating <newline>, but this can be changed by using the **RS** built-in variable. Each record of
5781 input shall be matched in turn against each pattern in the program. For each pattern matched,
5782 the associated action shall be executed.

The *awk* utility shall interpret each input record as a sequence of fields where, by default, a field is a string of non-<blank>s. This default white-space field delimiter can be changed by using the **FS** built-in variable or **-F ERE**. The *awk* utility shall denote the first field in a record \$1, the second \$2, and so on. The symbol \$0 shall refer to the entire record; setting any other field causes the re-evaluation of \$0. Assigning to \$0 shall reset the values of all other fields and the **NF** built-in variable.

5789 OPTIONS

5790 The `awk` utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
5791 12.2, Utility Syntax Guidelines.

5792 The following options shall be supported:

5793 -F ERE Define the input field separator to be the extended regular expression *ERE*, before any input is read; see **Regular Expressions** (on page 161).

5795 **-f** *progfile* Specify the pathname of the file *progfile* containing an *awk* program. If multiple
5796 instances of this option are specified, the concatenation of the files specified as
5797 *progfile* in the order specified shall be the *awk* program. The *awk* program can
5798 alternatively be specified in the command line as a single argument.

5799 -v *assignment*

5800 The application shall ensure that the *assignment* argument is in the same form as an
5801 *assignment* operand. The specified variable assignment shall occur prior to
5802 executing the awk program, including the actions associated with **BEGIN** patterns
5803 (if any). Multiple occurrences of this option can be specified.

5804 OPERANDS

5805 The following operands shall be supported:

5806 *program* If no **-f** option is specified, the first operand to *awk* shall be the text of the *awk*
5807 program. The application shall supply the *program* operand as a single argument to
5808 *awk*. If the text does not end in a <newline>, *awk* shall interpret the text as if it did.

5809 ***argument*** Either of the following two types of *argument* can be intermixed:

5810 *file* A pathname of a file that contains the input to be read, which is
5811 matched against the set of patterns in the program. If no *file* operands
5812 are specified, or if a *file* operand is ‘-’, the standard input shall be
5813 used.

5814 **assignment** An operand that begins with an underscore or alphabetic character
5815 from the portable character set (see the table in the Base Definitions
5816 volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set),
5817 followed by a sequence of underscores, digits, and alphabets from
5818 the portable character set, followed by the '=' character, shall
5819 specify a variable assignment rather than a pathname. The
5820 characters before the '=' represent the name of an *awk* variable; if
5821 that name is an *awk* reserved word (see **Grammar** (on page 170)) the
5822 behavior is undefined. The characters following the equal sign shall
5823 be interpreted as if they appeared in the *awk* program preceded and
5824 followed by a double-quote ('") character, as a **STRING** token (see
5825 **Grammar** (on page 170)), except that if the last character is an
5826 unescaped backslash, it shall be interpreted as a literal backslash
5827 rather than as the first character of the sequence "\ ". The variable
5828 shall be assigned the value of that **STRING** token and, if
5829 appropriate, shall be considered a *numeric string* (see **Expressions in**
5830 **awk** (on page 156)), the variable shall also be assigned its numeric
5831 value. Each such variable assignment shall occur just prior to the
5832 processing of the following *file*, if any. Thus, an assignment before
5833 the first *file* argument shall be executed after the **BEGIN** actions (if
5834 any), while an assignment after the last *file* argument shall occur
5835 before the **END** actions (if any). If there are no *file* arguments,
5836 assignments shall be executed before processing the standard input.

5837 **STDIN**

5838 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-';
5839 see the INPUT FILES section. If the *awk* program contains no actions and no patterns, but is
5840 otherwise a valid *awk* program, standard input and any *file* operands shall not be read and *awk*
5841 shall exit with a return status of zero.

5842 **INPUT FILES**

5843 Input files to the *awk* program from any of the following sources shall be text files:

- 5844 • Any *file* operands or their equivalents, achieved by modifying the *awk* variables **ARGV** and
5845 **ARGC**
- 5846 • Standard input in the absence of any *file* operands
- 5847 • Arguments to the **getline** function

5848 Whether the variable **RS** is set to a value other than a <newline> or not, for these files,
5849 implementations shall support records terminated with the specified separator up to
5850 {LINE_MAX} bytes and may support longer records.

5851 If **-f progfile** is specified, the application shall ensure that the files named by each of the *progfile*
5852 option-arguments are text files and their concatenation, in the same order as they appear in the
5853 arguments, is an *awk* program.

5854 **ENVIRONMENT VARIABLES**

5855 The following environment variables shall affect the execution of *awk*:

5856 **LANG** Provide a default value for the internationalization variables that are unset or null.
5857 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
5858 Internationalization Variables for the precedence of internationalization variables
5859 used to determine the values of locale categories.)

5860	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
5862	<i>LC_COLLATE</i>	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions and in comparisons of string values.
5866	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), the behavior of character classes within regular expressions, the identification of characters as letters, and the mapping of uppercase and lowercase characters for the toupper and tolower functions.
5871	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
5874	<i>LC_NUMERIC</i>	Determine the radix character used when interpreting numeric input, performing conversions between numeric and string values, and formatting numeric output. Regardless of locale, the period character (the decimal-point character of the POSIX locale) is the decimal-point character recognized in processing <i>awk</i> programs (including assignments in command line arguments).
5880 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
5881	<i>PATH</i>	Determine the search path when looking for commands executed by <i>system(expr)</i> , or input and output pipes; see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
5884	In addition, all environment variables shall be visible via the <i>awk</i> variable ENVIRON .	
5885	ASYNCHRONOUS EVENTS	
5886	Default.	
5887	STDOUT	The nature of the output files depends on the <i>awk</i> program.
5890	STDERR	The standard error shall be used only for diagnostic messages.
5891	OUTPUT FILES	The nature of the output files depends on the <i>awk</i> program.
5893	EXTENDED DESCRIPTION	
5894	Overall Program Structure	
5895	An <i>awk</i> program is composed of pairs of the form:	
5896	<i>pattern { action }</i>	
5897	Either the pattern or the action (including the enclosing brace characters) can be omitted.	
5898	A missing pattern shall match any record of input, and a missing action shall be equivalent to:	
5899	<i>{ print }</i>	
5900	Execution of the <i>awk</i> program shall start by first executing the actions associated with all BEGIN patterns in the order they occur in the program. Then each <i>file</i> operand (or standard input if no	

files were specified) shall be processed in turn by reading data from the file until a record separator is seen (<newline> by default). Before the first reference to a field in the record is evaluated, the record shall be split into fields, according to the rules in **Regular Expressions** (on page 161), using the value of **FS** that was current at the time the record was read. Each pattern in the program then shall be evaluated in the order of occurrence, and the action associated with each pattern that matches the current record executed. The action for a matching pattern shall be executed before evaluating subsequent patterns. Finally, the actions associated with all **END** patterns shall be executed in the order they occur in the program.

Expressions in awk

Expressions describe computations used in *patterns* and *actions*. In the following table, valid expression operations are given in groups from highest precedence first to lowest precedence last, with equal-precedence operators grouped between horizontal lines. In expression evaluation, where the grammar is formally ambiguous, higher precedence operators shall be evaluated before lower precedence operators. In this table *expr*, *expr1*, *expr2*, and *expr3* represent any expression, while *lvalue* represents any entity that can be assigned to (that is, on the left side of an assignment operator). The precise syntax of expressions is given in **Grammar** (on page 170).

Table 4-1 Expressions in Decreasing Precedence in *awk*

Syntax	Name	Type of Result	Associativity
(<i>expr</i>)	Grouping	Type of <i>expr</i>	N/A
\$ <i>expr</i>	Field reference	String	N/A
<i>lvalue</i> ++	Pre-increment	Numeric	N/A
<i>lvalue</i> --	Pre-decrement	Numeric	N/A
<i>lvalue</i> ++	Post-increment	Numeric	N/A
<i>lvalue</i> --	Post-decrement	Numeric	N/A
<i>expr</i> ^ <i>expr</i>	Exponentiation	Numeric	Right
! <i>expr</i>	Logical not	Numeric	N/A
+ <i>expr</i>	Unary plus	Numeric	N/A
- <i>expr</i>	Unary minus	Numeric	N/A
<i>expr</i> * <i>expr</i>	Multiplication	Numeric	Left
<i>expr</i> / <i>expr</i>	Division	Numeric	Left
<i>expr</i> % <i>expr</i>	Modulus	Numeric	Left
<i>expr</i> + <i>expr</i>	Addition	Numeric	Left
<i>expr</i> - <i>expr</i>	Subtraction	Numeric	Left
<i>expr</i> <i>expr</i>	String concatenation	String	Left
<i>expr</i> < <i>expr</i>	Less than	Numeric	None
<i>expr</i> <= <i>expr</i>	Less than or equal to	Numeric	None
<i>expr</i> != <i>expr</i>	Not equal to	Numeric	None
<i>expr</i> == <i>expr</i>	Equal to	Numeric	None
<i>expr</i> > <i>expr</i>	Greater than	Numeric	None
<i>expr</i> >= <i>expr</i>	Greater than or equal to	Numeric	None

	Syntax	Name	Type of Result	Associativity
5944	<i>expr</i> ~ <i>expr</i>	ERE match	Numeric	None
5945	<i>expr</i> !~ <i>expr</i>	ERE non-match	Numeric	None
5946	<i>expr</i> in array	Array membership	Numeric	Left
5947	(<i>index</i>) in array	Multi-dimension array membership	Numeric	Left
5948				
5949	<i>expr</i> && <i>expr</i>	Logical AND	Numeric	Left
5950	<i>expr</i> <i>expr</i>	Logical OR	Numeric	Left
5951	<i>expr1</i> ? <i>expr2</i> : <i>expr3</i>	Conditional expression	Type of selected <i>expr2</i> or <i>expr3</i>	Right
5952				
5953	<i>lvalue</i> ^= <i>expr</i>	Exponentiation assignment	Numeric	Right
5954	<i>lvalue</i> %= <i>expr</i>	Modulus assignment	Numeric	Right
5955	<i>lvalue</i> *= <i>expr</i>	Multiplication assignment	Numeric	Right
5956	<i>lvalue</i> /= <i>expr</i>	Division assignment	Numeric	Right
5957	<i>lvalue</i> += <i>expr</i>	Addition assignment	Numeric	Right
5958	<i>lvalue</i> -= <i>expr</i>	Subtraction assignment	Numeric	Right
5959	<i>lvalue</i> = <i>expr</i>	Assignment	Type of <i>expr</i>	Right
5960				
5961				

5962 Each expression shall have either a string value, a numeric value, or both. Except as stated for
 5963 specific contexts, the value of an expression shall be implicitly converted to the type needed for
 5964 the context in which it is used. A string value shall be converted to a numeric value by the
 5965 equivalent of the following calls to functions defined by the ISO C standard:

```
5966 setlocale(LC_NUMERIC, "") ;
5967 numeric_value = atof(string_value) ;
```

5968 A numeric value that is exactly equal to the value of an integer (see Section 1.7.2 (on page 7))
 5969 shall be converted to a string by the equivalent of a call to the **sprintf** function (see **String**
 5970 **Functions** (on page 167)) with the string "%d" as the *fmt* argument and the numeric value being
 5971 converted as the first and only *expr* argument. Any other numeric value shall be converted to a
 5972 string by the equivalent of a call to the **sprintf** function with the value of the variable
 5973 **CONVFMT** as the *fmt* argument and the numeric value being converted as the first and only
 5974 *expr* argument. The result of the conversion is unspecified if the value of **CONVFMT** is not a
 5975 floating-point format specification. This volume of IEEE Std 1003.1-2001 specifies no explicit
 5976 conversions between numbers and strings. An application can force an expression to be treated
 5977 as a number by adding zero to it, or can force it to be treated as a string by concatenating the null
 5978 string (" ") to it.

5979 A string value shall be considered a *numeric string* if it comes from one of the following:

- 5980 1. Field variables
- 5981 2. Input from the **getline()** function
- 5982 3. **FILENAME**
- 5983 4. **ARGV** array elements
- 5984 5. **ENVIRON** array elements
- 5985 6. Array elements created by the **split()** function
- 5986 7. A command line variable assignment

5987 8. Variable assignment from another numeric string variable

5988 and after all the following conversions have been applied, the resulting string would lexically be
5989 recognized as a **NUMBER** token as described by the lexical conventions in **Grammar** (on page
5990 170):

- 5991 • All leading and trailing <blank>s are discarded.
- 5992 • If the first non-<blank> is '+' or '-', it is discarded.
- 5993 • Changing each occurrence of the decimal point character from the current locale to a period.

5994 If a '-' character is ignored in the preceding description, the numeric value of the *numeric string*
5995 shall be the negation of the numeric value of the recognized **NUMBER** token. Otherwise, the
5996 numeric value of the *numeric string* shall be the numeric value of the recognized **NUMBER**
5997 token. Whether or not a string is a *numeric string* shall be relevant only in contexts where that
5998 term is used in this section.

5999 When an expression is used in a Boolean context, if it has a numeric value, a value of zero shall
6000 be treated as false and any other value shall be treated as true. Otherwise, a string value of the
6001 null string shall be treated as false and any other value shall be treated as true. A Boolean
6002 context shall be one of the following:

- 6003 • The first subexpression of a conditional expression
- 6004 • An expression operated on by logical NOT, logical AND, or logical OR
- 6005 • The second expression of a **for** statement
- 6006 • The expression of an **if** statement
- 6007 • The expression of the **while** clause in either a **while** or **do...while** statement
- 6008 • An expression used as a pattern (as in Overall Program Structure)

6009 All arithmetic shall follow the semantics of floating-point arithmetic as specified by the ISO C
6010 standard (see Section 1.7.2 (on page 7)).

6011 The value of the expression:

6012 *expr1* ^ *expr2*

6013 shall be equivalent to the value returned by the ISO C standard function call:

6014 *pow(expr1, expr2)*

6015 The expression:

6016 *lvalue* ^= *expr*

6017 shall be equivalent to the ISO C standard expression:

6018 *lvalue* = *pow(lvalue, expr)*

6019 except that *lvalue* shall be evaluated only once. The value of the expression:

6020 *expr1* % *expr2*

6021 shall be equivalent to the value returned by the ISO C standard function call:

6022 *fmod(expr1, expr2)*

6023 The expression:

6024 *lvalue* %= *expr*

6025 shall be equivalent to the ISO C standard expression:

6026 `lvalue = fmod(lvalue, expr)`

6027 except that lvalue shall be evaluated only once.

6028 Variables and fields shall be set by the assignment statement:

6029 `lvalue = expression`

6030 and the type of *expression* shall determine the resulting variable type. The assignment includes
6031 the arithmetic assignments ("`+=`", "`-=`", "`*=`", "`/=`", "`%=`", "`^=`", "`++`", "`--`") all of which
6032 shall produce a numeric result. The left-hand side of an assignment and the target of increment
6033 and decrement operators can be one of a variable, an array with index, or a field selector.

6034 The *awk* language supplies arrays that are used for storing numbers or strings. Arrays need not
6035 be declared. They shall initially be empty, and their sizes shall change dynamically. The
6036 subscripts, or element identifiers, are strings, providing a type of associative array capability. An
6037 array name followed by a subscript within square brackets can be used as an lvalue and thus as
6038 an expression, as described in the grammar; see **Grammar** (on page 170). Unsubscripted array
6039 names can be used in only the following contexts:

- A parameter in a function definition or function call
- The **NAME** token following any use of the keyword **in** as specified in the grammar (see
Grammar (on page 170)); if the name used in this context is not an array name, the behavior
is undefined

6044 A valid array *index* shall consist of one or more comma-separated expressions, similar to the way
6045 in which multi-dimensional arrays are indexed in some programming languages. Because *awk*
6046 arrays are really one-dimensional, such a comma-separated list shall be converted to a single
6047 string by concatenating the string values of the separate expressions, each separated from the
6048 other by the value of the **SUBSEP** variable. Thus, the following two index operations shall be
6049 equivalent:

6050 `var[expr1, expr2, ... exprn]`

6051 `var[expr1 SUBSEP expr2 SUBSEP ... SUBSEP exprn]`

6052 The application shall ensure that a multi-dimensioned *index* used with the **in** operator is
6053 parenthesized. The **in** operator, which tests for the existence of a particular array element, shall
6054 not cause that element to exist. Any other reference to a nonexistent array element shall
6055 automatically create it.

6056 Comparisons (with the '`<`', '`<=`', '`!=`', '`==`', '`>`', and '`>=`' operators) shall be made
6057 numerically if both operands are numeric, if one is numeric and the other has a string value that
6058 is a numeric string, or if one is numeric and the other has the uninitialized value. Otherwise,
6059 operands shall be converted to strings as required and a string comparison shall be made using
6060 the locale-specific collation sequence. The value of the comparison expression shall be 1 if the
6061 relation is true, or 0 if the relation is false.

6062

Variables and Special Variables

6063

Variables can be used in an *awk* program by referencing them. With the exception of function parameters (see **User-Defined Functions** (on page 169)), they are not explicitly declared. Function parameter names shall be local to the function; all other variable names shall be global. The same name shall not be used as both a function parameter name and as the name of a function or a special *awk* variable. The same name shall not be used both as a variable name with global scope and as the name of a function. The same name shall not be used within the same scope both as a scalar variable and as an array. Uninitialized variables, including scalar variables, array elements, and field variables, shall have an uninitialized value. An uninitialized value shall have both a numeric value of zero and a string value of the empty string. Evaluation of variables with an uninitialized value, to either string or numeric, shall be determined by the context in which they are used.

6064

Field variables shall be designated by a '\$' followed by a number or numerical expression. The effect of the field number *expression* evaluating to anything other than a non-negative integer is unspecified; uninitialized variables or string values need not be converted to numeric values in this context. New field variables can be created by assigning a value to them. References to nonexistent fields (that is, fields after **SNF**), shall evaluate to the uninitialized value. Such references shall not create new fields. However, assigning to a nonexistent field (for example, $\$({NF+2})=5$) shall increase the value of **NF**; create any intervening fields with the uninitialized value; and cause the value of **\$0** to be recomputed, with the fields being separated by the value of **OFS**. Each field variable shall have a string value or an uninitialized value when created. Field variables shall have the uninitialized value when created from **\$0** using **FS** and the variable does not contain any characters. If appropriate, the field variable shall be considered a numeric string (see **Expressions in awk** (on page 156)).

6065

6066

6067

6068

6069

6070

6071

6072

6073

6074

6075

6076

6077

6078

6079

6080

6081

6082

6083

6084

6085

6086

Implementations shall support the following other special variables that are set by *awk*:

6087

ARGC The number of elements in the **ARGV** array.

6088

ARGV An array of command line arguments, excluding options and the *program* argument, numbered from zero to **ARGC**-1.

6090

The arguments in **ARGV** can be modified or added to; **ARGC** can be altered. As each input file ends, *awk* shall treat the next non-null element of **ARGV**, up to the current value of **ARGC**-1, inclusive, as the name of the next input file. Thus, setting an element of **ARGV** to null means that it shall not be treated as an input file. The name '-' indicates the standard input. If an argument matches the format of an *assignment* operand, this argument shall be treated as an *assignment* rather than a *file* argument.

6091

6092

6093

6094

6095

6096

CONVFMT The **printf** format for converting numbers to strings (except for output statements, where **OFMT** is used); "% .6g" by default.

6097

ENVIRON An array representing the value of the environment, as described in the **exec** functions defined in the System Interfaces volume of IEEE Std 1003.1-2001. The indices of the array shall be strings consisting of the names of the environment variables, and the value of each array element shall be a string consisting of the value of that variable. If appropriate, the environment variable shall be considered a *numeric string* (see **Expressions in awk** (on page 156)); the array element shall also have its numeric value.

6098

6099

6100

6101

6102

6103

6104

6105

6106

In all cases where the behavior of *awk* is affected by environment variables (including the environment of any commands that *awk* executes via the **system** function or via pipeline redirections with the **print** statement, the **printf** statement, or the **getline** function), the environment used shall be the environment at the time

6107

6108

6109

6110	6111	awk began executing; it is implementation-defined whether any modification of ENVIRON affects this environment.
6112	6113	FILENAME A pathname of the current input file. Inside a BEGIN action the value is undefined. Inside an END action the value shall be the name of the last input file processed.
6115	FNR	The ordinal number of the current record in the current file. Inside a BEGIN action the value shall be zero. Inside an END action the value shall be the number of the last record processed in the last file processed.
6118	FS	Input field separator regular expression; a <space> by default.
6119	NF	The number of fields in the current record. Inside a BEGIN action, the use of NF is undefined unless a getline function without a <i>var</i> argument is executed previously. Inside an END action, NF shall retain the value it had for the last record read, unless a subsequent, redirected, getline function without a <i>var</i> argument is performed prior to entering the END action.
6124	NR	The ordinal number of the current record from the start of input. Inside a BEGIN action the value shall be zero. Inside an END action the value shall be the number of the last record processed.
6127	OFMT	The printf format for converting numbers to strings in output statements (see Output Statements (on page 165)); "%.6g" by default. The result of the conversion is unspecified if the value of OFMT is not a floating-point format specification.
6131	OFS	The print statement output field separation; <space> by default.
6132	ORS	The print statement output record separator; a <newline> by default.
6133	RLENGTH	The length of the string matched by the match function.
6134	RS	The first character of the string value of RS shall be the input record separator; a <newline> by default. If RS contains more than one character, the results are unspecified. If RS is null, then records are separated by sequences consisting of a <newline> plus one or more blank lines, leading or trailing blank lines shall not result in empty records at the beginning or end of the input, and a <newline> shall always be a field separator, no matter what the value of FS is.
6140	RSTART	The starting position of the string matched by the match function, numbering from 1. This shall always be equivalent to the return value of the match function.
6142	SUBSEP	The subscript separator string for multi-dimensional arrays; the default value is implementation-defined.

6144 Regular Expressions

6145 The *awk* utility shall make use of the extended regular expression notation (see the Base
6146 Definitions volume of IEEE Std 1003.1-2001, Section 9.4, Extended Regular Expressions) except
6147 that it shall allow the use of C-language conventions for escaping special characters within the
6148 EREs, as specified in the table in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5,
6149 File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v') and the following
6150 table; these escape sequences shall be recognized both inside and outside bracket expressions.
6151 Note that records need not be separated by <newline>s and string constants can contain
6152 <newline>s, so even the "\n" sequence is valid in *awk* EREs. Using a slash character within an
6153 ERE requires the escaping shown in the following table.

6154

Table 4-2 Escape Sequences in awk

Escape Sequence	Description	Meaning
\"	Backslash quotation-mark	Quotation-mark character
\/	Backslash slash	Slash character
\ddd	A backslash character followed by the longest sequence of one, two, or three octal-digit characters (01234567). If all of the digits are 0 (that is, representation of the NUL character), the behavior is undefined.	The character whose encoding is represented by the one, two, or three-digit octal integer. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading '\' for each byte.
\c	A backslash character followed by any character not described in this table or in the table in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v').	Undefined

6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172 A regular expression can be matched against a specific field or string by using one of the two regular expression matching operators, '`~`' and '`!~`'. These operators shall interpret their right-hand operand as a regular expression and their left-hand operand as a string. If the regular expression matches the string, the '`~`' expression shall evaluate to a value of 1, and the '`!~`' expression shall evaluate to a value of 0. (The regular expression matching operation is as defined by the term matched in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.1, Regular Expression Definitions, where a match occurs on any part of the string unless the regular expression is limited with the circumflex or dollar sign special characters.) If the regular expression does not match the string, the '`~`' expression shall evaluate to a value of 0, and the '`!~`' expression shall evaluate to a value of 1. If the right-hand operand is any expression other than the lexical token ERE, the string value of the expression shall be interpreted as an extended regular expression, including the escape conventions described above. Note that these same escape conventions shall also be applied in determining the value of a string literal (the lexical token STRING), and thus shall be applied a second time when a string literal is used in this context.

6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186 When an ERE token appears as an expression in any context other than as the right-hand of the '`~`' or '`!~`' operator or as one of the built-in function arguments described below, the value of the resulting expression shall be the equivalent of:

6187 \$0 ~ /ere/

6188 The ere argument to the **gsub**, **match**, **sub** functions, and the fs argument to the **split** function (see **String Functions** (on page 167)) shall be interpreted as extended regular expressions. These can be either ERE tokens or arbitrary expressions, and shall be interpreted in the same manner as the right-hand side of the '`~`' or '`!~`' operator.

6189 An extended regular expression can be used to separate fields by using the **-F ERE** option or by 6190 assigning a string containing the expression to the built-in variable **FS**. The default value of the 6191 **FS** variable shall be a single <space>. The following describes **FS** behavior:

6192 1. If **FS** is a null string, the behavior is unspecified.

- 6200 2. If **FS** is a single character:
- 6201 a. If **FS** is <space>, skip leading and trailing <blank>s; fields shall be delimited by sets
6202 of one or more <blank>s.
- 6203 b. Otherwise, if **FS** is any other character *c*, fields shall be delimited by each single
6204 occurrence of *c*.
- 6205 3. Otherwise, the string value of **FS** shall be considered to be an extended regular expression.
6206 Each occurrence of a sequence matching the extended regular expression shall delimit
6207 fields.

6208 Except for the ' ~ ' and " ! ~ " operators, and in the **gsub**, **match**, **split**, and **sub** built-in functions,
6209 ERE matching shall be based on input records; that is, record separator characters (the first
6210 character of the value of the variable **RS**, <newline> by default) cannot be embedded in the
6211 expression, and no expression shall match the record separator character. If the record separator
6212 is not <newline>, <newline>s embedded in the expression can be matched. For the ' ~ ' and
6213 " ! ~ " operators, and in those four built-in functions, ERE matching shall be based on text strings;
6214 that is, any character (including <newline> and the record separator) can be embedded in the
6215 pattern, and an appropriate pattern shall match any character. However, in all *awk* ERE
6216 matching, the use of one or more NUL characters in the pattern, input record, or text string
6217 produces undefined results.

6218 Patterns

6219 A *pattern* is any valid *expression*, a range specified by two expressions separated by a comma, or
6220 one of the two special patterns **BEGIN** or **END**.

6221 Special Patterns

6222 The *awk* utility shall recognize two special patterns, **BEGIN** and **END**. Each **BEGIN** pattern
6223 shall be matched once and its associated action executed before the first record of input is read
6224 (except possibly by use of the **getline** function—see **Input/Output and General Functions** (on
6225 page 168)—in a prior **BEGIN** action) and before command line assignment is done. Each **END**
6226 pattern shall be matched once and its associated action executed after the last record of input has
6227 been read. These two patterns shall have associated actions.

6228 **BEGIN** and **END** shall not combine with other patterns. Multiple **BEGIN** and **END** patterns
6229 shall be allowed. The actions associated with the **BEGIN** patterns shall be executed in the order
6230 specified in the program, as are the **END** actions. An **END** pattern can precede a **BEGIN** pattern
6231 in a program.

6232 If an *awk* program consists of only actions with the pattern **BEGIN**, and the **BEGIN** action
6233 contains no **getline** function, *awk* shall exit without reading its input when the last statement in
6234 the last **BEGIN** action is executed. If an *awk* program consists of only actions with the pattern
6235 **END** or only actions with the patterns **BEGIN** and **END**, the input shall be read before the
6236 statements in the **END** actions are executed.

6237 **Expression Patterns**

6238 An expression pattern shall be evaluated as if it were an expression in a Boolean context. If the
6239 result is true, the pattern shall be considered to match, and the associated action (if any) shall be
6240 executed. If the result is false, the action shall not be executed.

6241 **Pattern Ranges**

6242 A pattern range consists of two expressions separated by a comma; in this case, the action shall
6243 be performed for all records between a match of the first expression and the following match of
6244 the second expression, inclusive. At this point, the pattern range can be repeated starting at
6245 input records subsequent to the end of the matched range.

6246 **Actions**

6247 An action is a sequence of statements as shown in the grammar in **Grammar** (on page 170). Any
6248 single statement can be replaced by a statement list enclosed in braces. The application shall
6249 ensure that statements in a statement list are separated by <newline>s or semicolons. Statements
6250 in a statement list shall be executed sequentially in the order that they appear.

6251 The **expression** acting as the conditional in an **if** statement shall be evaluated and if it is non-zero
6252 or non-null, the following statement shall be executed; otherwise, if **else** is present, the statement
6253 following the **else** shall be executed.

6254 The **if**, **while**, **do...while**, **for**, **break**, and **continue** statements are based on the ISO C standard
6255 (see Section 1.7.2 (on page 7)), except that the Boolean expressions shall be treated as described
6256 in **Expressions in awk** (on page 156), and except in the case of:

6257 `for (variable in array)`

6258 which shall iterate, assigning each **index** of **array** to **variable** in an unspecified order. The results of
6259 adding new elements to **array** within such a **for** loop are undefined. If a **break** or **continue**
6260 statement occurs outside of a loop, the behavior is undefined.

6261 The **delete** statement shall remove an individual array element. Thus, the following code deletes
6262 an entire array:

6263 `for (index in array)
6264 delete array[index]`

6265 The **next** statement shall cause all further processing of the current input record to be
6266 abandoned. The behavior is undefined if a **next** statement appears or is invoked in a **BEGIN** or
6267 **END** action.

6268 The **exit** statement shall invoke all **END** actions in the order in which they occur in the program
6269 source and then terminate the program without reading further input. An **exit** statement inside
6270 an **END** action shall terminate the program without further execution of **END** actions. If an
6271 expression is specified in an **exit** statement, its numeric value shall be the exit status of **awk**,
6272 unless subsequent errors are encountered or a subsequent **exit** statement with an expression is
6273 executed.

6274 **Output Statements**

6275 Both **print** and **printf** statements shall write to standard output by default. The output shall be
6276 written to the location specified by *output redirection* if one is supplied, as follows:

6277 > *expression*
6278 >> *expression*
6279 | *expression*

6280 In all cases, the *expression* shall be evaluated to produce a string that is used as a pathname into
6281 which to write (for '>' or ">>") or as a command to be executed (for '|'). Using the first two
6282 forms, if the file of that name is not currently open, it shall be opened, creating it if necessary and
6283 using the first form, truncating the file. The output then shall be appended to the file. As long as
6284 the file remains open, subsequent calls in which *expression* evaluates to the same string value
6285 shall simply append output to the file. The file remains open until the **close** function (see
6286 **Input/Output and General Functions** (on page 168)) is called with an expression that evaluates
6287 to the same string value.

6288 The third form shall write output onto a stream piped to the input of a command. The stream
6289 shall be created if no stream is currently open with the value of *expression* as its command name.
6290 The stream created shall be equivalent to one created by a call to the *popen()* function defined in
6291 the System Interfaces volume of IEEE Std 1003.1-2001 with the value of *expression* as the
6292 *command* argument and a value of *w* as the *mode* argument. As long as the stream remains open,
6293 subsequent calls in which *expression* evaluates to the same string value shall write output to the
6294 existing stream. The stream shall remain open until the **close** function (see **Input/Output and**
6295 **General Functions** (on page 168)) is called with an expression that evaluates to the same string
6296 value. At that time, the stream shall be closed as if by a call to the *pclose()* function defined in
6297 the System Interfaces volume of IEEE Std 1003.1-2001.

6298 As described in detail by the grammar in **Grammar** (on page 170), these output statements shall
6299 take a comma-separated list of *expressions* referred to in the grammar by the non-terminal
6300 symbols **expr_list**, **print_expr_list**, or **print_expr_list_opt**. This list is referred to here as the
6301 *expression list*, and each member is referred to as an *expression argument*.

6302 The **print** statement shall write the value of each *expression argument* onto the indicated output
6303 stream separated by the current output field separator (see variable **OFS** above), and terminated
6304 by the output record separator (see variable **ORS** above). All *expression arguments* shall be
6305 taken as strings, being converted if necessary; this conversion shall be as described in
6306 **Expressions in awk** (on page 156), with the exception that the **printf** format in **OFMT** shall be
6307 used instead of the value in **CONVFMT**. An empty *expression list* shall stand for the whole
6308 input record (\$0).

6309 The **printf** statement shall produce output based on a notation similar to the File Format
6310 Notation used to describe file formats in this volume of IEEE Std 1003.1-2001 (see the Base
6311 Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation). Output shall be
6312 produced as specified with the first *expression argument* as the string *format* and subsequent
6313 *expression arguments* as the strings *arg1* to *argn*, inclusive, with the following exceptions:

- 6314 1. The *format* shall be an actual character string rather than a graphical representation.
6315 Therefore, it cannot contain empty character positions. The <space> in the *format* string, in
6316 any context other than a *flag* of a conversion specification, shall be treated as an ordinary
6317 character that is copied to the output.
- 6318 2. If the character set contains a 'Δ' character and that character appears in the *format* string,
6319 it shall be treated as an ordinary character that is copied to the output.

- 6320 3. The *escape sequences* beginning with a backslash character shall be treated as sequences of
6321 ordinary characters that are copied to the output. Note that these same sequences shall be
6322 interpreted lexically by *awk* when they appear in literal strings, but they shall not be
6323 treated specially by the **printf** statement.
- 6324 4. A *field width* or *precision* can be specified as the '*' character instead of a digit string. In
6325 this case the next argument from the expression list shall be fetched and its numeric value
6326 taken as the field width or precision.
- 6327 5. The implementation shall not precede or follow output from the d or u conversion
6328 specifier characters with <blank>s not specified by the *format* string.
- 6329 6. The implementation shall not precede output from the o conversion specifier character
6330 with leading zeros not specified by the *format* string.
- 6331 7. For the c conversion specifier character: if the argument has a numeric value, the character
6332 whose encoding is that value shall be output. If the value is zero or is not the encoding of
6333 any character in the character set, the behavior is undefined. If the argument does not have
6334 a numeric value, the first character of the string value shall be output; if the string does not
6335 contain any characters, the behavior is undefined.
- 6336 8. For each conversion specification that consumes an argument, the next expression
6337 argument shall be evaluated. With the exception of the c conversion specifier character,
6338 the value shall be converted (according to the rules specified in **Expressions in awk** (on
6339 page 156)) to the appropriate type for the conversion specification.
- 6340 9. If there are insufficient expression arguments to satisfy all the conversion specifications in
6341 the *format* string, the behavior is undefined.
- 6342 10. If any character sequence in the *format* string begins with a '%' character, but does not
6343 form a valid conversion specification, the behavior is unspecified.

6344 Both **print** and **printf** can output at least {LINE_MAX} bytes.

6345 Functions

6346 The *awk* language has a variety of built-in functions: arithmetic, string, input/output, and
6347 general.

6348 Arithmetic Functions

6349 The arithmetic functions, except for **int**, shall be based on the ISO C standard (see Section 1.7.2
6350 (on page 7)). The behavior is undefined in cases where the ISO C standard specifies that an error
6351 be returned or that the behavior is undefined. Although the grammar (see **Grammar** (on page
6352 170)) permits built-in functions to appear with no arguments or parentheses, unless the
6353 argument or parentheses are indicated as optional in the following list (by displaying them
6354 within the "[]" brackets), such use is undefined.

- 6355 **atan2(y,x)** Return arctangent of y/x in radians in the range [-π,π].
6356 **cos(x)** Return cosine of x, where x is in radians.
6357 **sin(x)** Return sine of x, where x is in radians.
6358 **exp(x)** Return the exponential function of x.
6359 **log(x)** Return the natural logarithm of x.
6360 **sqrt(x)** Return the square root of x.

6361 **int(x)** Return the argument truncated to an integer. Truncation shall be toward 0 when
 6362 x>0.
 6363 **rand()** Return a random number *n*, such that $0 \leq n < 1$.
 6364 **srand([expr])** Set the seed value for *rand* to *expr* or use the time of day if *expr* is omitted. The
 6365 previous seed value shall be returned.

6366 String Functions

The string functions in the following list shall be supported. Although the grammar (see **Grammar** (on page 170)) permits built-in functions to appear with no arguments or parentheses, unless the argument or parentheses are indicated as optional in the following list (by displaying them within the " [] " brackets), such use is undefined.

6371 **gsub(ere, repl[], in)**
 6372 Behave like **sub** (see below), except that it shall replace all occurrences of the
 6373 regular expression (like the *ed* utility global substitute) in \$0 or in the *in* argument,
 6374 when specified.
 6375 **index(s, t)** Return the position, in characters, numbering from 1, in string *s* where string *t* first
 6376 occurs, or zero if it does not occur at all.
 6377 **length([s])** Return the length, in characters, of its argument taken as a string, or of the whole
 6378 record, \$0, if there is no argument.
 6379 **match(s, ere)** Return the position, in characters, numbering from 1, in string *s* where the
 6380 extended regular expression *ere* occurs, or zero if it does not occur at all. RSTART
 6381 shall be set to the starting position (which is the same as the returned value), zero
 6382 if no match is found; RLENGTH shall be set to the length of the matched string, -1
 6383 if no match is found.
 6384 **split(s, a[, fs])**
 6385 Split the string *s* into array elements *a*[1], *a*[2], ..., *a*[*n*], and return *n*. All elements
 6386 of the array shall be deleted before the split is performed. The separation shall be
 6387 done with the ERE *fs* or with the field separator FS if *fs* is not given. Each array
 6388 element shall have a string value when created and, if appropriate, the array
 6389 element shall be considered a numeric string (see **Expressions in awk** (on page
 6390 156)). The effect of a null string as the value of *fs* is unspecified.
 6391 **sprintf(fmt, expr, expr, ...)**
 6392 Format the expressions according to the **printf** format given by *fmt* and return the
 6393 resulting string.
 6394 **sub(ere, repl[], in])**
 6395 Substitute the string *repl* in place of the first instance of the extended regular
 6396 expression *ERE* in string *in* and return the number of substitutions. An ampersand
 6397 (' & ') appearing in the string *repl* shall be replaced by the string from *in* that
 6398 matches the *ERE*. An ampersand preceded with a backslash ('\') shall be
 6399 interpreted as the literal ampersand character. An occurrence of two consecutive
 6400 backslashes shall be interpreted as just a single literal backslash character. Any
 6401 other occurrence of a backslash (for example, preceding any other character) shall
 6402 be treated as a literal backslash character. Note that if *repl* is a string literal (the
 6403 lexical token **STRING**; see **Grammar** (on page 170)), the handling of the
 6404 ampersand character occurs after any lexical processing, including any lexical
 6405 backslash escape sequence processing. If *in* is specified and it is not an lvalue (see
 6406 **Expressions in awk** (on page 156)), the behavior is undefined. If *in* is omitted, awk

6407		shall use the current record (\$0) in its place.
6408	substr(s, m[, n])	Return the at most <i>n</i> -character substring of <i>s</i> that begins at position <i>m</i> , numbering from 1. If <i>n</i> is omitted, or if <i>n</i> specifies more characters than are left in the string, the length of the substring shall be limited by the length of the string <i>s</i> .
6409		
6410		
6411		
6412	tolower(s)	Return a string based on the string <i>s</i> . Each character in <i>s</i> that is an uppercase letter specified to have a tolower mapping by the <i>LC_CTYPE</i> category of the current locale shall be replaced in the returned string by the lowercase letter specified by the mapping. Other characters in <i>s</i> shall be unchanged in the returned string.
6413		
6414		
6415		
6416	toupper(s)	Return a string based on the string <i>s</i> . Each character in <i>s</i> that is a lowercase letter specified to have a toupper mapping by the <i>LC_CTYPE</i> category of the current locale is replaced in the returned string by the uppercase letter specified by the mapping. Other characters in <i>s</i> are unchanged in the returned string.
6417		
6418		
6419		
6420	All of the preceding functions that take <i>ERE</i> as a parameter expect a pattern or a string valued expression that is a regular expression as defined in Regular Expressions (on page 161).	
6421		
6422	Input/Output and General Functions	
6423	The input/output and general functions are:	
6424	close(expression)	Close the file or pipe opened by a print or printf statement or a call to getline with the same string-valued <i>expression</i> . The limit on the number of open <i>expression</i> arguments is implementation-defined. If the close was successful, the function shall return zero; otherwise, it shall return non-zero.
6425		
6426		
6427		
6428		
6429	expression getline [var]	Read a record of input from a stream piped from the output of a command. The stream shall be created if no stream is currently open with the value of <i>expression</i> as its command name. The stream created shall be equivalent to one created by a call to the <i>popen()</i> function with the value of <i>expression</i> as the <i>command</i> argument and a value of <i>r</i> as the <i>mode</i> argument. As long as the stream remains open, subsequent calls in which <i>expression</i> evaluates to the same string value shall read subsequent records from the stream. The stream shall remain open until the close function is called with an expression that evaluates to the same string value. At that time, the stream shall be closed as if by a call to the <i>pclose()</i> function. If <i>var</i> is omitted, \$0 and NF shall be set; otherwise, <i>var</i> shall be set and, if appropriate, it shall be considered a numeric string (see Expressions in awk (on page 156)).
6430		
6431		
6432		
6433		
6434		
6435		
6436		
6437		
6438		
6439		
6440		
6441	The getline operator can form ambiguous constructs when there are unparenthesized operators (including concatenate) to the left of the ' ' (to the beginning of the expression containing getline). In the context of the '\$' operator, ' ' shall behave as if it had a lower precedence than '\$'. The result of evaluating other operators is unspecified, and conforming applications shall parenthesize properly all such usages.	
6442		
6443		
6444		
6445		
6446		
6447	getline	Set \$0 to the next input record from the current input file. This form of getline shall set the NF , NR , and FNR variables.
6448		
6449	getline var	Set variable <i>var</i> to the next input record from the current input file and, if appropriate, <i>var</i> shall be considered a numeric string (see Expressions in awk (on page 156)). This form of getline shall set the FNR and NR variables.
6450		
6451		

6452 `getline [var] < expression`

6453 Read the next record of input from a named file. The *expression* shall be evaluated
6454 to produce a string that is used as a pathname. If the file of that name is not
6455 currently open, it shall be opened. As long as the stream remains open, subsequent
6456 calls in which *expression* evaluates to the same string value shall read subsequent
6457 records from the file. The file shall remain open until the **close** function is called
6458 with an expression that evaluates to the same string value. If *var* is omitted, \$0 and
6459 *NF* shall be set; otherwise, *var* shall be set and, if appropriate, it shall be considered
6460 a numeric string (see **Expressions in awk** (on page 156)).

6461 The **getline** operator can form ambiguous constructs when there are
6462 unparenthesized binary operators (including concatenate) to the right of the '<'
6463 (up to the end of the expression containing the **getline**). The result of evaluating
6464 such a construct is unspecified, and conforming applications shall parenthesize
6465 properly all such usages.

6466 `system(expression)`

6467 Execute the command given by *expression* in a manner equivalent to the **system()**
6468 function defined in the System Interfaces volume of IEEE Std 1003.1-2001 and
6469 return the exit status of the command.

6470 All forms of **getline** shall return 1 for successful input, zero for end-of-file, and -1 for an error.

6471 Where strings are used as the name of a file or pipeline, the application shall ensure that the
6472 strings are textually identical. The terminology "same string value" implies that "equivalent
6473 strings", even those that differ only by <space>s, represent different files.

6474 User-Defined Functions

6475 The *awk* language also provides user-defined functions. Such functions can be defined as:

```
6476      function name([parameter, . . .]) { statements }
```

6477 A function can be referred to anywhere in an *awk* program; in particular, its use can precede its
6478 definition. The scope of a function is global.

6479 Function parameters, if present, can be either scalars or arrays; the behavior is undefined if an
6480 array name is passed as a parameter that the function uses as a scalar, or if a scalar expression is
6481 passed as a parameter that the function uses as an array. Function parameters shall be passed by
6482 value if scalar and by reference if array name.

6483 The number of parameters in the function definition need not match the number of parameters
6484 in the function call. Excess formal parameters can be used as local variables. If fewer arguments
6485 are supplied in a function call than are in the function definition, the extra parameters that are
6486 used in the function body as scalars shall evaluate to the uninitialized value until they are
6487 otherwise initialized, and the extra parameters that are used in the function body as arrays shall
6488 be treated as uninitialized arrays where each element evaluates to the uninitialized value until
6489 otherwise initialized.

6490 When invoking a function, no white space can be placed between the function name and the
6491 opening parenthesis. Function calls can be nested and recursive calls can be made upon
6492 functions. Upon return from any nested or recursive function call, the values of all of the calling
6493 function's parameters shall be unchanged, except for array parameters passed by reference. The
6494 **return** statement can be used to return a value. If a **return** statement appears outside of a
6495 function definition, the behavior is undefined.

6496 In the function definition, <newline>s shall be optional before the opening brace and after the
6497 closing brace. Function definitions can appear anywhere in the program where a *pattern-action*

6498 pair is allowed.

6499 Grammar

6500 The grammar in this section and the lexical conventions in the following section shall together
 6501 describe the syntax for *awk* programs. The general conventions for this style of grammar are
 6502 described in Section 1.10 (on page 19). A valid program can be represented as the non-terminal
 6503 symbol *program* in the grammar. This formal syntax shall take precedence over the preceding
 6504 text syntax description.

```

6505 %token NAME NUMBER STRING ERE
6506 %token FUNC_NAME /* Name followed by '()' without white space. */
6507 /* Keywords */
6508 %token Begin End
6509 /* 'BEGIN' 'END' */
6510 %token Break Continue Delete Do Else
6511 /* 'break' 'continue' 'delete' 'do' 'else' */
6512 %token Exit For Function If In
6513 /* 'exit' 'for' 'function' 'if' 'in' */
6514 %token Next Print Printf Return While
6515 /* 'next' 'print' 'printf' 'return' 'while' */
6516 /* Reserved function names */
6517 %token BUILTIN_FUNC_NAME
6518 /* One token for the following:
   * atan2 cos sin exp log sqrt int rand srand
   * gsub index length match split sprintf sub
   * substr tolower toupper close system
   */
6519 %token GETLINE
6520 /* Syntactically different from other built-ins. */
6521 /* Two-character tokens. */
6522 %token ADD_ASSIGN SUB_ASSIGN MUL_ASSIGN DIV_ASSIGN MOD_ASSIGN POW_ASSIGN
6523 /* '+=' '-=' '*=' '/=' '%=' '^=' */
6524 /* OR AND NO_MATCH EQ LE GE NE INCR DECR APPEND */
6525 /* '|'| '&&' '!~' '===' '<=' '>=' '!=' '++' '--' '>>' */
6526 /* One-character tokens. */
6527 %token '{' '}' '(' ')' '[' ']' ',' ';' NEWLINE
6528 %token '+' '-' '*' '%' '^' '!' '>' '<' '|' '?' ':' '^' '$' '='
6529 /*start program
6530 %%*/
6531 program : item_list
6532 | actionless_item_list
6533 ;
6534
6535 item_list : newline_opt
6536 | actionless_item_list item_terminator
6537 | item_list item_terminator
6538 | item_list action_terminator
6539 ;
6540
6541
6542

```

```
6543     actionless_item_list : item_list           pattern terminator
6544         | actionless_item_list pattern terminator
6545         ;
6546     item       : pattern action
6547         | Function NAME      '(' param_list_opt ')'
6548             newline_opt action
6549         | Function FUNC_NAME '(' param_list_opt ')'
6550             newline_opt action
6551         ;
6552     param_list_opt : /* empty */
6553         | param_list
6554         ;
6555     param_list   : NAME
6556         | param_list ',' NAME
6557         ;
6558     pattern     : Begin
6559         | End
6560         | expr
6561         | expr ',' newline_opt expr
6562         ;
6563     action      : '{' newline_opt           '}'
6564         | '{' newline_opt terminated_statement_list  '}'
6565         | '{' newline_opt unterminated_statement_list '}'
6566         ;
6567     terminator  : terminator ';'          '
6568         | terminator NEWLINE
6569         |
6570             ';'          '
6571             NEWLINE
6572     terminated_statement_list : terminated_statement
6573         | terminated_statement_list terminated_statement
6574         ;
6575     unterminated_statement_list : unterminated_statement
6576         | terminated_statement_list unterminated_statement
6577         ;
6578     terminated_statement : action newline_opt
6579         | If '(' expr ')' newline_opt terminated_statement
6580         | If '(' expr ')' newline_opt terminated_statement
6581             Else newline_opt terminated_statement
6582         | While '(' expr ')' newline_opt terminated_statement
6583         | For '(' simple_statement_opt ';' '
6584             expr_opt ';' simple_statement_opt ')' newline_opt
6585             terminated_statement
6586         | For '(' NAME In NAME ')' newline_opt
6587             terminated_statement
6588         | ';' newline_opt
6589         | terminatable_statement NEWLINE newline_opt
6590         | terminatable_statement ';'     newline_opt
```

```
6591          ;
6592      unterminated_statement : terminatable_statement
6593          | If '(' expr ')' newline_opt unterminated_statement
6594          | If '(' expr ')' newline_opt terminated_statement
6595              Else newline_opt unterminated_statement
6596          | While '(' expr ')' newline_opt unterminated_statement
6597          | For '(' simple_statement_opt ';' '
6598              expr_opt ';' simple_statement_opt ')' newline_opt
6599                  unterminated_statement
6600          | For '(' NAME In NAME ')' newline_opt
6601                  unterminated_statement
6602          ;
6603      terminatable_statement : simple_statement
6604          | Break
6605          | Continue
6606          | Next
6607          | Exit expr_opt
6608          | Return expr_opt
6609          | Do newline_opt terminated_statement While '(' expr ')'
6610          ;
6611      simple_statement_opt : /* empty */
6612          | simple_statement
6613          ;
6614      simple_statement : Delete NAME '[' expr_list ']'
6615          | expr
6616          | print_statement
6617          ;
6618      print_statement : simple_print_statement
6619          | simple_print_statement output_redirection
6620          ;
6621      simple_print_statement : Print print_expr_list_opt
6622          | Print '(' multiple_expr_list ')'
6623          | Printf print_expr_list
6624          | Printf '(' multiple_expr_list ')'
6625          ;
6626      output_redirection : '>'     expr
6627          | APPEND expr
6628          | '| '    expr
6629          ;
6630      expr_list_opt : /* empty */
6631          | expr_list
6632          ;
6633      expr_list : expr
6634          | multiple_expr_list
6635          ;
6636      multiple_expr_list : expr ',' newline_opt expr
6637          | multiple_expr_list ',' newline_opt expr
```

```

6638          ;
6639      expr_opt   : /* empty */
6640          | expr
6641          ;
6642      expr       : unary_expr
6643          | non_unary_expr
6644          ;
6645      unary_expr : '+' expr
6646          | '-' expr
6647          | unary_expr '^' expr
6648          | unary_expr '*' expr
6649          | unary_expr '/' expr
6650          | unary_expr '%' expr
6651          | unary_expr '+' expr
6652          | unary_expr '-' expr
6653          | unary_expr non_unary_expr
6654          | unary_expr '<' expr
6655          | unary_expr LE expr
6656          | unary_expr NE expr
6657          | unary_expr EQ expr
6658          | unary_expr '>' expr
6659          | unary_expr GE expr
6660          | unary_expr '^' expr
6661          | unary_expr NO_MATCH expr
6662          | unary_expr In NAME
6663          | unary_expr AND newline_opt expr
6664          | unary_expr OR newline_opt expr
6665          | unary_expr '?' expr ':' expr
6666          | unary_input_function
6667          ;
6668      non_unary_expr : '(' expr ')'
6669          | '!' expr
6670          | non_unary_expr '^' expr
6671          | non_unary_expr '*' expr
6672          | non_unary_expr '/' expr
6673          | non_unary_expr '%' expr
6674          | non_unary_expr '+' expr
6675          | non_unary_expr '-' expr
6676          | non_unary_expr non_unary_expr
6677          | non_unary_expr '<' expr
6678          | non_unary_expr LE expr
6679          | non_unary_expr NE expr
6680          | non_unary_expr EQ expr
6681          | non_unary_expr '>' expr
6682          | non_unary_expr GE expr
6683          | non_unary_expr '^' expr
6684          | non_unary_expr NO_MATCH expr
6685          | non_unary_expr In NAME
6686          | '(' multiple_expr_list ')' In NAME
6687          | non_unary_expr AND newline_opt expr

```

```
6688      | non_unary_expr OR  newline_opt expr
6689      | non_unary_expr '?' expr ':' expr
6690      | NUMBER
6691      | STRING
6692      | lvalue
6693      | ERE
6694      | lvalue INCR
6695      | lvalue DECR
6696      | INCR lvalue
6697      | DECR lvalue
6698      | lvalue POW_ASSIGN expr
6699      | lvalue MOD_ASSIGN expr
6700      | lvalue MUL_ASSIGN expr
6701      | lvalue DIV_ASSIGN expr
6702      | lvalue ADD_ASSIGN expr
6703      | lvalue SUB_ASSIGN expr
6704      | lvalue '=' expr
6705      | FUNC_NAME '(' expr_list_opt ')'
6706          /* no white space allowed before '(' */
6707      | BUILTIN_FUNC_NAME '(' expr_list_opt ')'
6708      | BUILTIN_FUNC_NAME
6709      | non_unary_input_function
6710      ;
6711 print_expr_list_opt : /* empty */
6712     | print_expr_list
6713     ;
6714 print_expr_list : print_expr
6715     | print_expr_list ',' newline_opt print_expr
6716     ;
6717 print_expr : unary_print_expr
6718     | non_unary_print_expr
6719     ;
6720 unary_print_expr : '+' print_expr
6721     | '-' print_expr
6722     | unary_print_expr '^' print_expr
6723     | unary_print_expr '*' print_expr
6724     | unary_print_expr '/' print_expr
6725     | unary_print_expr '%' print_expr
6726     | unary_print_expr '+' print_expr
6727     | unary_print_expr '-' print_expr
6728     | unary_print_expr non_unary_print_expr
6729     | unary_print_expr '~~' print_expr
6730     | unary_print_expr NO_MATCH print_expr
6731     | unary_print_expr In NAME
6732     | unary_print_expr AND newline_opt print_expr
6733     | unary_print_expr OR  newline_opt print_expr
6734     | unary_print_expr '?' print_expr ':' print_expr
6735     ;
6736 non_unary_print_expr : '(' expr ')'
6737     | '!' print_expr
```

```

6738      | non_unary_print_expr '^'      print_expr
6739      | non_unary_print_expr '*'      print_expr
6740      | non_unary_print_expr '/'      print_expr
6741      | non_unary_print_expr '%'     print_expr
6742      | non_unary_print_expr '+'      print_expr
6743      | non_unary_print_expr '-'      print_expr
6744      | non_unary_print_expr         non_unary_print_expr
6745      | non_unary_print_expr '~~'     print_expr
6746      | non_unary_print_expr NO_MATCH print_expr
6747      | non_unary_print_expr In NAME
6748      | (' multiple_expr_list ') In NAME
6749      | non_unary_print_expr AND newline_opt print_expr
6750      | non_unary_print_expr OR  newline_opt print_expr
6751      | non_unary_print_expr '?' print_expr ':' print_expr
6752      | NUMBER
6753      | STRING
6754      | lvalue
6755      | ERE
6756      | lvalue INCR
6757      | lvalue DECR
6758      | INCR lvalue
6759      | DECR lvalue
6760      | lvalue POW_ASSIGN print_expr
6761      | lvalue MOD_ASSIGN print_expr
6762      | lvalue MUL_ASSIGN print_expr
6763      | lvalue DIV_ASSIGN print_expr
6764      | lvalue ADD_ASSIGN print_expr
6765      | lvalue SUB_ASSIGN print_expr
6766      | lvalue '=' print_expr
6767      | FUNC_NAME '(' expr_list_opt ')'
6768          /* no white space allowed before '(' */
6769      | BUILTIN_FUNC_NAME '(' expr_list_opt ')'
6770      | BUILTIN_FUNC_NAME
6771      ;
6772      lvalue      : NAME
6773              | NAME '[' expr_list ']'
6774              | '$' expr
6775      ;
6776      non_unary_input_function : simple_get
6777          | simple_get '<' expr
6778          | non_unary_expr '| simple_get
6779      ;
6780      unary_input_function : unary_expr '| simple_get
6781          ;
6782      simple_get      : GETLINE
6783          | GETLINE lvalue
6784      ;
6785      newline_opt      : /* empty */
6786          | newline_opt NEWLINE
6787      ;

```

6788 This grammar has several ambiguities that shall be resolved as follows:

- 6789 • Operator precedence and associativity shall be as described in Table 4-1 (on page 156).
- 6790 • In case of ambiguity, an **else** shall be associated with the most immediately preceding **if** that
6791 would satisfy the grammar.
- 6792 • In some contexts, a slash (' / ') that is used to surround an ERE could also be the division
6793 operator. This shall be resolved in such a way that wherever the division operator could
6794 appear, a slash is assumed to be the division operator. (There is no unary division operator.)

6795 One convention that might not be obvious from the formal grammar is where <newline>s are
6796 acceptable. There are several obvious placements such as terminating a statement, and a
6797 backslash can be used to escape <newline>s between any lexical tokens. In addition, <newline>s
6798 without backslashes can follow a comma, an open brace, logical AND operator ("&&"), logical
6799 OR operator ("||"), the **do** keyword, the **else** keyword, and the closing parenthesis of an **if**, **for**,
6800 or **while** statement. For example:

6801 { print \$1,
6802 \$2 }

6803 Lexical Conventions

6804 The lexical conventions for *awk* programs, with respect to the preceding grammar, shall be as
6805 follows:

- 6806 1. Except as noted, *awk* shall recognize the longest possible token or delimiter beginning at a
6807 given point.
- 6808 2. A comment shall consist of any characters beginning with the number sign character and
6809 terminated by, but excluding the next occurrence of, a <newline>. Comments shall have
6810 no effect, except to delimit lexical tokens.
- 6811 3. The <newline> shall be recognized as the token **NEWLINE**.
- 6812 4. A backslash character immediately followed by a <newline> shall have no effect.
- 6813 5. The token **STRING** shall represent a string constant. A string constant shall begin with the
6814 character '\"'. Within a string constant, a backslash character shall be considered to begin
6815 an escape sequence as specified in the table in the Base Definitions volume of
6816 IEEE Std 1003.1-2001, Chapter 5, File Format Notation ('\\", '\a', '\b', '\f', '\n',
6817 '\r', '\t', '\v'). In addition, the escape sequences in Table 4-2 (on page 162) shall be
6818 recognized. A <newline> shall not occur within a string constant. A string constant shall be
6819 terminated by the first unescaped occurrence of the character '\"' after the one that begins
6820 the string constant. The value of the string shall be the sequence of all unescaped
6821 characters and values of escape sequences between, but not including, the two delimiting
6822 '\"' characters.
- 6823 6. The token **ERE** represents an extended regular expression constant. An ERE constant shall
6824 begin with the slash character. Within an ERE constant, a backslash character shall be
6825 considered to begin an escape sequence as specified in the table in the Base Definitions
6826 volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation. In addition, the escape
6827 sequences in Table 4-2 (on page 162) shall be recognized. The application shall ensure that
6828 a <newline> does not occur within an ERE constant. An ERE constant shall be terminated
6829 by the first unescaped occurrence of the slash character after the one that begins the ERE
6830 constant. The extended regular expression represented by the ERE constant shall be the
6831 sequence of all unescaped characters and values of escape sequences between, but not
6832 including, the two delimiting slash characters.

- 6833 7. A <blank> shall have no effect, except to delimit lexical tokens or within **STRING** or **ERE**
 6834 tokens.
- 6835 8. The token **NUMBER** shall represent a numeric constant. Its form and numeric value shall
 6836 be equivalent to either of the tokens **floating-constant** or **integer-constant** as specified by
 6837 the ISO C standard, with the following exceptions:
- 6838 a. An integer constant cannot begin with 0x or include the hexadecimal digits 'a', 'b',
 6839 'c', 'd', 'e', 'f', 'A', 'B', 'C', 'D', 'E', or 'F'.
- 6840 b. The value of an integer constant beginning with 0 shall be taken in decimal rather
 6841 than octal.
- 6842 c. An integer constant cannot include a suffix ('u', 'U', 'l', or 'L').
- 6843 d. A floating constant cannot include a suffix ('f', 'F', 'l', or 'L').

6844 If the value is too large or too small to be representable (see Section 1.7.2 (on page 7)), the
 6845 behavior is undefined.

- 6846 9. A sequence of underscores, digits, and alphabetics from the portable character set (see the
 6847 Base Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set),
 6848 beginning with an underscore or alphabetic, shall be considered a word.
- 6849 10. The following words are keywords that shall be recognized as individual tokens; the name
 6850 of the token is the same as the keyword:

BEGIN	delete	END	function	in	printf
break	do	exit	getline	next	return
continue	else	for	if	print	while

- 6854 11. The following words are names of built-in functions and shall be recognized as the token
 6855 **BUILTIN_FUNC_NAME**:

atan2	gsub	log	split	sub	toupper
close	index	match	sprintf	substr	
cos	int	rand	sqrt	system	
exp	length	sin	srand	tolower	

6860 The above-listed keywords and names of built-in functions are considered reserved words.

- 6861 12. The token **NAME** shall consist of a word that is not a keyword or a name of a built-in
 6862 function and is not followed immediately (without any delimiters) by the '(' character.
- 6863 13. The token **FUNC_NAME** shall consist of a word that is not a keyword or a name of a
 6864 built-in function, followed immediately (without any delimiters) by the '(' character. The
 6865 ')' character shall not be included as part of the token.
- 6866 14. The following two-character sequences shall be recognized as the named tokens:

Token Name	Sequence	Token Name	Sequence
ADD_ASSIGN	+=	NO_MATCH	!~
SUB_ASSIGN	-=	EQ	==
MUL_ASSIGN	*=	LE	<=
DIV_ASSIGN	/=	GE	>=
MOD_ASSIGN	%=	NE	!=
POW_ASSIGN	^=	INCR	++
OR		DECR	--
AND	&&	APPEND	>>

6876 15. The following single characters shall be recognized as tokens whose names are the
6877 character:

6878 <newline> { } () [] , ; + - * % ^ ! > < | ? : ~ \$ =

6879 There is a lexical ambiguity between the token **ERE** and the tokens **'/'** and **DIV_ASSIGN**.
6880 When an input sequence begins with a slash character in any syntactic context where the token
6881 **'/'** or **DIV_ASSIGN** could appear as the next token in a valid program, the longer of those two
6882 tokens that can be recognized shall be recognized. In any other syntactic context where the token
6883 **ERE** could appear as the next token in a valid program, the token **ERE** shall be recognized.

6884 EXIT STATUS

6885 The following exit values shall be returned:

6886 0 All input files were processed successfully.

6887 >0 An error occurred.

6888 The exit status can be altered within the program by using an **exit** expression.

6889 CONSEQUENCES OF ERRORS

6890 If any *file* operand is specified and the named file cannot be accessed, *awk* shall write a
6891 diagnostic message to standard error and terminate without any further action.

6892 If the program specified by either the *program* operand or a *progfile* operand is not a valid *awk*
6893 program (as specified in the EXTENDED DESCRIPTION section), the behavior is undefined.

6894 APPLICATION USAGE

6895 The **index**, **length**, **match**, and **substr** functions should not be confused with similar functions in
6896 the ISO C standard; the *awk* versions deal with characters, while the ISO C standard deals with
6897 bytes.

6898 Because the concatenation operation is represented by adjacent expressions rather than an
6899 explicit operator, it is often necessary to use parentheses to enforce the proper evaluation
6900 precedence.

6901 EXAMPLES

6902 The *awk* program specified in the command line is most easily specified within single-quotes (for
6903 example, '*program*') for applications using *sh*, because *awk* programs commonly contain
6904 characters that are special to the shell, including double-quotes. In the cases where an *awk*
6905 program contains single-quote characters, it is usually easiest to specify most of the program as
6906 strings within single-quotes concatenated by the shell with quoted single-quote characters. For
6907 example:

6908 awk '''\'' { print "quote:", \$0 }'

6909 prints all lines from the standard input containing a single-quote character, prefixed with *quote*:

6910 The following are examples of simple *awk* programs:

6911 1. Write to the standard output all input lines for which field 3 is greater than 5:

6912 \$3 > 5

6913 2. Write every tenth line:

6914 (NR % 10) == 0

6915 3. Write any line with a substring matching the regular expression:

6916 /(G|D)(2[0-9][[:alpha:]])*/

- 6917 4. Print any line with a substring containing a 'G' or 'D', followed by a sequence of digits
 6918 and characters. This example uses character classes **digit** and **alpha** to match language-
 6919 independent digit and alphabetic characters respectively:
- ```
6920 / (G|D) ([[[:digit:]][:alpha:]]*) /
```
- 6921        5. Write any line in which the second field matches the regular expression and the fourth  
 6922        field does not:
- ```
6923 $2 ~ /xyz/ && $4 !~ /xyz/
```
- 6924 6. Write any line in which the second field contains a backslash:
- ```
6925 $2 ~ /\\\/
```
- 6926        7. Write any line in which the second field contains a backslash. Note that backslash escapes  
 6927        are interpreted twice; once in lexical processing of the string and once in processing the  
 6928        regular expression:
- ```
6929 $2 ~ "\\\\\"
```
- 6930 8. Write the second to the last and the last field in each line. Separate the fields by a colon:
- ```
6931 {OFS=":";print $(NF-1), $NF}
```
- 6932        9. Write the line number and number of fields in each line. The three strings representing the  
 6933        line number, the colon, and the number of fields are concatenated and that string is written  
 6934        to standard output:
- ```
6935 {print NR ":" NF}
```
- 6936 10. Write lines longer than 72 characters:
- ```
6937 length($0) > 72
```
- 6938        11. Write the first two fields in opposite order separated by **OFS**:
- ```
6939 { print $2, $1 }
```
- 6940 12. Same, with input fields separated by a comma or <space>s and <tab>s, or both:
- ```
6941 BEGIN { FS = "[\t]*|[\t]+"; }
6942 { print $2, $1 }
```
- 6943        13. Add up the first column, print sum, and average:
- ```
6944 { s += $1 }  
6945 END {print "sum is ", s, " average is", s/NR}
```
- 6946 14. Write fields in reverse order, one per line (many lines out for each line in):
- ```
6947 { for (i = NF; i > 0; --i) print $i }
```
- 6948        15. Write all lines between occurrences of the strings **start** and **stop**:
- ```
6949 /start/, /stop/
```
- 6950 16. Write all lines whose first field is different from the previous one:
- ```
6951 $1 != prev { print; prev = $1 }
```
- 6952        17. Simulate **echo**:
- ```
6953 BEGIN {  
6954   for (i = 1; i < ARGC; ++i)  
6955     printf("%s%s", ARGV[i], i==ARGC-1?"\n":")
```

6956 }

6957 18. Write the path prefixes contained in the *PATH* environment variable, one per line:

```
6958            BEGIN {  
6959              n = split (ENVIRON["PATH"], path, ":")  
6960              for (i = 1; i <= n; ++i)  
6961                print path[i]  
6962            }
```

6963 19. If there is a file named **input** containing page headers of the form:

6964 Page #

6965 and a file named **program** that contains:

```
6966            /Page/ { $2 = n++; }  
6967            { print }
```

6968 then the command line:

6969 awk -f program n=5 input

6970 prints the file **input**, filling in page numbers starting at 5.

6971 RATIONALE

6972 This description is based on the new *awk*, “nawk”, (see the referenced *The AWK Programming*
6973 *Language*), which introduced a number of new features to the historical *awk*:

- 6974 1. New keywords: **delete**, **do**, **function**, **return**
- 6975 2. New built-in functions: **atan2**, **close**, **cos**, **gsub**, **match**, **rand**, **sin**, **srand**, **sub**, **system**
- 6976 3. New predefined variables: **FNR**, **ARGC**, **ARGV**, **RSTART**, **RLENGTH**, **SUBSEP**
- 6977 4. New expression operators: ?, :, , ^
- 6978 5. The **FS** variable and the third argument to **split**, now treated as extended regular
6979 expressions.
- 6980 6. The operator precedence, changed to more closely match the C language. Two examples
6981 of code that operate differently are:

```
6982            while ( n /= 10 > 1) ...  
6983            if (!"wk" ~ /bwk/) ...
```

6984 Several features have been added based on newer implementations of *awk*:

- 6985 • Multiple instances of **-f** *progfile* are permitted.
- 6986 • The new option **-v** *assignment*.
- 6987 • The new predefined variable **ENVIRON**.
- 6988 • New built-in functions **toupper** and **tolower**.
- 6989 • More formatting capabilities are added to **printf** to match the ISO C standard.

6990 The overall *awk* syntax has always been based on the C language, with a few features from the
6991 shell command language and other sources. Because of this, it is not completely compatible with
6992 any other language, which has caused confusion for some users. It is not the intent of the
6993 standard developers to address such issues. A few relatively minor changes toward making the
6994 language more compatible with the ISO C standard were made; most of these changes are based
6995 on similar changes in recent implementations, as described above. There remain several C-

language conventions that are not in *awk*. One of the notable ones is the comma operator, which is commonly used to specify multiple expressions in the C language **for** statement. Also, there are various places where *awk* is more restrictive than the C language regarding the type of expression that can be used in a given context. These limitations are due to the different features that the *awk* language does provide.

Regular expressions in *awk* have been extended somewhat from historical implementations to make them a pure superset of extended regular expressions, as defined by IEEE Std 1003.1-2001 (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.4, Extended Regular Expressions). The main extensions are internationalization features and interval expressions. Historical implementations of *awk* have long supported backslash escape sequences as an extension to extended regular expressions, and this extension has been retained despite inconsistency with other utilities. The number of escape sequences recognized in both extended regular expressions and strings has varied (generally increasing with time) among implementations. The set specified by IEEE Std 1003.1-2001 includes most sequences known to be supported by popular implementations and by the ISO C standard. One sequence that is not supported is hexadecimal value escapes beginning with '\x'. This would allow values expressed in more than 9 bits to be used within *awk* as in the ISO C standard. However, because this syntax has a non-deterministic length, it does not permit the subsequent character to be a hexadecimal digit. This limitation can be dealt with in the C language by the use of lexical string concatenation. In the *awk* language, concatenation could also be a solution for strings, but not for extended regular expressions (either lexical ERE tokens or strings used dynamically as regular expressions). Because of this limitation, the feature has not been added to IEEE Std 1003.1-2001.

When a string variable is used in a context where an extended regular expression normally appears (where the lexical token ERE is used in the grammar) the string does not contain the literal slashes.

Some versions of *awk* allow the form:

```
func name(args, ... ) { statements }
```

This has been deprecated by the authors of the language, who asked that it not be specified.

Historical implementations of *awk* produce an error if a **next** statement is executed in a **BEGIN** action, and cause *awk* to terminate if a **next** statement is executed in an **END** action. This behavior has not been documented, and it was not believed that it was necessary to standardize it.

The specification of conversions between string and numeric values is much more detailed than in the documentation of historical implementations or in the referenced *The AWK Programming Language*. Although most of the behavior is designed to be intuitive, the details are necessary to ensure compatible behavior from different implementations. This is especially important in relational expressions since the types of the operands determine whether a string or numeric comparison is performed. From the perspective of an application writer, it is usually sufficient to expect intuitive behavior and to force conversions (by adding zero or concatenating a null string) when the type of an expression does not obviously match what is needed. The intent has been to specify historical practice in almost all cases. The one exception is that, in historical implementations, variables and constants maintain both string and numeric values after their original value is converted by any use. This means that referencing a variable or constant can have unexpected side effects. For example, with historical implementations the following program:

```
{  
    a = "+2"  
    b = 2
```

```

7044     if (NR % 2)
7045         c = a + b
7046     if (a == b)
7047         print "numeric comparison"
7048     else
7049         print "string comparison"
7050 }
```

7051 would perform a numeric comparison (and output numeric comparison) for each odd-numbered line, but perform a string comparison (and output string comparison) for each even-numbered line. IEEE Std 1003.1-2001 ensures that comparisons will be numeric if necessary. With historical implementations, the following program:

```

7055 BEGIN {
7056     OFMT = "%e"
7057     print 3.14
7058     OFMT = "%f"
7059     print 3.14
7060 }
```

7061 would output "3.140000e+00" twice, because in the second **print** statement the constant "3.14" would have a string value from the previous conversion. IEEE Std 1003.1-2001 requires that the output of the second **print** statement be "3.14000". The behavior of historical implementations was seen as too unintuitive and unpredictable.

7065 It was pointed out that with the rules contained in early drafts, the following script would print nothing:

```

7067 BEGIN {
7068     y[1.5] = 1
7069     OFMT = "%e"
7070     print y[1.5]
7071 }
```

7072 Therefore, a new variable, **CONVFMT**, was introduced. The **OFMT** variable is now restricted to 7073 affecting output conversions of numbers to strings and **CONVFMT** is used for internal 7074 conversions, such as comparisons or array indexing. The default value is the same as that for 7075 **OFMT**, so unless a program changes **CONVFMT** (which no historical program would do), it 7076 will receive the historical behavior associated with internal string conversions.

7077 The POSIX *awk* lexical and syntactic conventions are specified more formally than in other 7078 sources. Again the intent has been to specify historical practice. One convention that may not be 7079 obvious from the formal grammar as in other verbal descriptions is where <newline>s are 7080 acceptable. There are several obvious placements such as terminating a statement, and a 7081 backslash can be used to escape <newline>s between any lexical tokens. In addition, <newline>s 7082 without backslashes can follow a comma, an open brace, a logical AND operator ("&&"), a 7083 logical OR operator ("| |"), the **do** keyword, the **else** keyword, and the closing parenthesis of an 7084 **if**, **for**, or **while** statement. For example:

```

7085 { print $1,
7086     $2 }
```

7087 The requirement that *awk* add a trailing <newline> to the program argument text is to simplify 7088 the grammar, making it match a text file in form. There is no way for an application or test suite 7089 to determine whether a literal <newline> is added or whether *awk* simply acts as if it did.

7090 IEEE Std 1003.1-2001 requires several changes from historical implementations in order to
7091 support internationalization. Probably the most subtle of these is the use of the decimal-point
7092 character, defined by the *LC_NUMERIC* category of the locale, in representations of floating-
7093 point numbers. This locale-specific character is used in recognizing numeric input, in converting
7094 between strings and numeric values, and in formatting output. However, regardless of locale,
7095 the period character (the decimal-point character of the POSIX locale) is the decimal-point
7096 character recognized in processing awk programs (including assignments in command line
7097 arguments). This is essentially the same convention as the one used in the ISO C standard. The
7098 difference is that the C language includes the *setlocale()* function, which permits an application
7099 to modify its locale. Because of this capability, a C application begins executing with its locale
7100 set to the C locale, and only executes in the environment-specified locale after an explicit call to
7101 *setlocale()*. However, adding such an elaborate new feature to the awk language was seen as
7102 inappropriate for IEEE Std 1003.1-2001. It is possible to execute an awk program explicitly in any
7103 desired locale by setting the environment in the shell.

7104 The undefined behavior resulting from NULs in extended regular expressions allows future
7105 extensions for the GNU gawk program to process binary data.

7106 The behavior in the case of invalid awk programs (including lexical, syntactic, and semantic
7107 errors) is undefined because it was considered overly limiting on implementations to specify. In
7108 most cases such errors can be expected to produce a diagnostic and a non-zero exit status.
7109 However, some implementations may choose to extend the language in ways that make use of
7110 certain invalid constructs. Other invalid constructs might be deemed worthy of a warning, but
7111 otherwise cause some reasonable behavior. Still other constructs may be very difficult to detect
7112 in some implementations. Also, different implementations might detect a given error during an
7113 initial parsing of the program (before reading any input files) while others might detect it when
7114 executing the program after reading some input. Implementors should be aware that diagnosing
7115 errors as early as possible and producing useful diagnostics can ease debugging of applications,
7116 and thus make an implementation more usable.

7117 The unspecified behavior from using multi-character RS values is to allow possible future
7118 extensions based on extended regular expressions used for record separators. Historical
7119 implementations take the first character of the string and ignore the others.

7120 Unspecified behavior when *split(string,array,<null>)* is used is to allow a proposed future
7121 extension that would split up a string into an array of individual characters.

7122 In the context of the **getline** function, equally good arguments for different precedences of the |
7123 and < operators can be made. Historical practice has been that:

7124 `getline < "a" "b"`

7125 is parsed as:

7126 `(getline < "a") "b"`

7127 although many would argue that the intent was that the file **ab** should be read. However:

7128 `getline < "x" + 1`

7129 parses as:

7130 `getline < ("x" + 1)`

7131 Similar problems occur with the | version of **getline**, particularly in combination with \$. For
7132 example:

7133 `$"echo hi" | getline`

7134 (This situation is particularly problematic when used in a **print** statement, where the **|getline**
7135 part might be a redirection of the **print**.)

7136 Since in most cases such constructs are not (or at least should not) be used (because they have a
7137 natural ambiguity for which there is no conventional parsing), the meaning of these constructs
7138 has been made explicitly unspecified. (The effect is that a conforming application that runs into
7139 the problem must parenthesize to resolve the ambiguity.) There appeared to be few if any actual
7140 uses of such constructs.

7141 Grammars can be written that would cause an error under these circumstances. Where
7142 backwards-compatibility is not a large consideration, implementors may wish to use such
7143 grammars.

7144 Some historical implementations have allowed some built-in functions to be called without an
7145 argument list, the result being a default argument list chosen in some “reasonable” way. Use of
7146 **length** as a synonym for **length(\$0)** is the only one of these forms that is thought to be widely
7147 known or widely used; this particular form is documented in various places (for example, most
7148 historical *awk* reference pages, although not in the referenced *The AWK Programming Language*)
7149 as legitimate practice. With this exception, default argument lists have always been
7150 undocumented and vaguely defined, and it is not at all clear how (or if) they should be
7151 generalized to user-defined functions. They add no useful functionality and preclude possible
7152 future extensions that might need to name functions without calling them. Not standardizing
7153 them seems the simplest course. The standard developers considered that **length** merited special
7154 treatment, however, since it has been documented in the past and sees possibly substantial use
7155 in historical programs. Accordingly, this usage has been made legitimate, but Issue 5 removed
7156 the obsolescent marking for XSI-conforming implementations and many otherwise conforming
7157 applications depend on this feature.

7158 In **sub** and **gsub**, if *repl* is a string literal (the lexical token **STRING**), then two consecutive
7159 backslash characters should be used in the string to ensure a single backslash will precede the
7160 ampersand when the resultant string is passed to the function. (For example, to specify one
7161 literal ampersand in the replacement string, use **gsub(ERE, "\\\&")**.)

7162 Historically the only special character in the *repl* argument of **sub** and **gsub** string functions was
7163 the ampersand ('&') character and preceding it with the backslash character was used to turn
7164 off its special meaning.

7165 The description in the ISO POSIX-2:1993 standard introduced behavior such that the backslash
7166 character was another special character and it was unspecified whether there were any other
7167 special characters. This description introduced several portability problems, some of which are
7168 described below, and so it has been replaced with the more historical description. Some of the
7169 problems include:

- 7170 • Historically, to create the replacement string, a script could use **gsub(ERE, "\\\&")**, but with
7171 the ISO POSIX-2:1993 standard wording, it was necessary to use **gsub(ERE, "\\\\\\\&")**.
7172 Backslash characters are doubled here because all string literals are subject to lexical analysis,
7173 which would reduce each pair of backslash characters to a single backslash before being
7174 passed to **gsub**.
- 7175 • Since it was unspecified what the special characters were, for portable scripts to guarantee
7176 that characters are printed literally, each character had to be preceded with a backslash. (For
7177 example, a portable script had to use **gsub(ERE, "\\\h\\\i")** to produce a replacement string
7178 of "hi".)

7179 The description for comparisons in the ISO POSIX-2:1993 standard did not properly describe
7180 historical practice because of the way numeric strings are compared as numbers. The current
7181 rules cause the following code:

```
7182     if (0 == "000")
7183         print "strange, but true"
7184     else
7185         print "not true"
```

7186 to do a numeric comparison, causing the `if` to succeed. It should be intuitively obvious that this
7187 is incorrect behavior, and indeed, no historical implementation of `awk` actually behaves this way.

7188 To fix this problem, the definition of *numeric string* was enhanced to include only those values
7189 obtained from specific circumstances (mostly external sources) where it is not possible to
7190 determine unambiguously whether the value is intended to be a string or a numeric.

7191 Variables that are assigned to a numeric string shall also be treated as a numeric string. (For
7192 example, the notion of a numeric string can be propagated across assignments.) In comparisons,
7193 all variables having the uninitialized value are to be treated as a numeric operand evaluating to
7194 the numeric value zero.

7195 Uninitialized variables include all types of variables including scalars, array elements, and fields.
7196 The definition of an uninitialized value in **Variables and Special Variables** (on page 160) is
7197 necessary to describe the value placed on uninitialized variables and on fields that are valid (for
7198 example, < \$NF) but have no characters in them and to describe how these variables are to be
7199 used in comparisons. A valid field, such as `$1`, that has no characters in it can be obtained from
7200 an input line of "\t\t" when `FS='\'\t'`. Historically, the comparison (`$1<10`) was done
7201 numerically after evaluating `$1` to the value zero.

7202 The phrase "... also shall have the numeric value of the numeric string" was removed from
7203 several sections of the ISO POSIX-2:1993 standard because is specifies an unnecessary
7204 implementation detail. It is not necessary for IEEE Std 1003.1-2001 to specify that these objects be
7205 assigned two different values. It is only necessary to specify that these objects may evaluate to
7206 two different values depending on context.

7207 The description of numeric string processing is based on the behavior of the `atof()` function in
7208 the ISO C standard. While it is not a requirement for an implementation to use this function,
7209 many historical implementations of `awk` do. In the ISO C standard, floating-point constants use a
7210 period as a decimal point character for the language itself, independent of the current locale, but
7211 the `atof()` function and the associated `strtod()` function use the decimal point character of the
7212 current locale when converting strings to numeric values. Similarly in `awk`, floating-point
7213 constants in an `awk` script use a period independent of the locale, but input strings use the
7214 decimal point character of the locale.

7215 FUTURE DIRECTIONS

7216 None.

7217 SEE ALSO

7218 Section 1.10 (on page 19), `grep`, `lex`, `sed`, the System Interfaces volume of IEEE Std 1003.1-2001,
7219 `atof()`, `exec`, `popen()`, `setlocale()`, `strtod()`

7220 CHANGE HISTORY

7221 First released in Issue 2.

7222 Issue 5

7223 The FUTURE DIRECTIONS section is added.

7224 Issue 6

7225 The `awk` utility is aligned with the IEEE P1003.2b draft standard.

7226 The normative text is reworded to avoid use of the term "must" for application requirements.

7227 IEEE PASC Interpretation 1003.2 #211 is applied, adding the sentence “An occurrence of two
7228 consecutive backslashes shall be interpreted as just a single literal backslash character.” into the
7229 description of the **sub** string function.

7230 NAME

7231 basename — return non-directory portion of a pathname

7232 SYNOPSIS

7233 basename *string* [*suffix*]

7234 DESCRIPTION

7235 The *string* operand shall be treated as a pathname, as defined in the Base Definitions volume of
7236 IEEE Std 1003.1-2001, Section 3.266, Pathname. The string *string* shall be converted to the
7237 filename corresponding to the last pathname component in *string* and then the suffix string
7238 *suffix*, if present, shall be removed. This shall be done by performing actions equivalent to the
7239 following steps in order:

- 7240 1. If *string* is a null string, it is unspecified whether the resulting string is ‘.’ or a null string.
7241 In either case, skip steps 2 through 6.
- 7242 2. If *string* is “//”, it is implementation-defined whether steps 3 to 6 are skipped or
7243 processed.
- 7244 3. If *string* consists entirely of slash characters, *string* shall be set to a single slash character. In
7245 this case, skip steps 4 to 6.
- 7246 4. If there are any trailing slash characters in *string*, they shall be removed.
- 7247 5. If there are any slash characters remaining in *string*, the prefix of *string* up to and including
7248 the last slash character in *string* shall be removed.
- 7249 6. If the *suffix* operand is present, is not identical to the characters remaining in *string*, and is
7250 identical to a suffix of the characters remaining in *string*, the suffix *suffix* shall be removed
7251 from *string*. Otherwise, *string* is not modified by this step. It shall not be considered an
7252 error if *suffix* is not found in *string*.

7253 The resulting string shall be written to standard output.

7254 OPTIONS

7255 None.

7256 OPERANDS

7257 The following operands shall be supported:

7258 *string* A string.
7259 *suffix* A string.

7260 STDIN

7261 Not used.

7262 INPUT FILES

7263 None.

7264 ENVIRONMENT VARIABLES

7265 The following environment variables shall affect the execution of *basename*:

7266 *LANG* Provide a default value for the internationalization variables that are unset or null.
7267 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
7268 Internationalization Variables for the precedence of internationalization variables
7269 used to determine the values of locale categories.)
7270 *LC_ALL* If set to a non-empty string value, override the values of all the other
7271 internationalization variables.

LC_MESSAGES Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

7278 XSI NLSPATH Determine the location of message catalogs for the processing of *LC_MESSAGES*.

7279 ASYNCHRONOUS EVENTS

7280 Default.

7281 STDOUT

7282 The *basename* utility shall write a line to the standard output in the following format:

7283 "%s\n", <resulting string>

7284 STDERR

7285 The standard error shall be used only for diagnostic messages.

7286 OUTPUT FILES

7287 None.

7288 EXTENDED DESCRIPTION

7289 None.

7290 EXIT STATUS

7291 The following exit values shall be returned:

7292 0 Successful completion.

7293 >0 An error occurred.

7294 CONSEQUENCES OF ERRORS

7295 Default.

7296 APPLICATION USAGE

7297 The definition of *pathname* specifies implementation-defined behavior for pathnames starting
7298 with two slash characters. Therefore, applications shall not arbitrarily add slashes to the
7299 beginning of a pathname unless they can ensure that there are more or less than two or are
7300 prepared to deal with the implementation-defined consequences.

7301 EXAMPLES

7302 If the string *string* is a valid pathname:

7303 \$(basename "string")

7304 produces a filename that could be used to open the file named by *string* in the directory returned
7305 by:

7306 \$(dirname "string")

If the string *string* is not a valid pathname, the same algorithm is used, but the result need not be a valid filename. The *basename* utility is not expected to make any judgements about the validity of *string* as a pathname; it just follows the specified algorithm to produce a result string.

7310 The following shell script compiles `/usr/src/cmd/cat.c` and moves the output to a file named `cat`
7311 in the current directory when invoked with the argument `/usr/src/cmd/cat` or with the argument
7312 `/usr/src/cmd/cat.c`:

7313 c99 \$(dirname "\$1")/\$(basename "\$1" .c).c
7314 mv a.out \$(basename "\$1" .c)

7315 **RATIONALE**

7316 The behaviors of *basename* and *dirname* have been coordinated so that when *string* is a valid
7317 pathname:

7318 \$(basename "string")

7319 would be a valid filename for the file in the directory:

7320 \$(dirname "string")

7321 This would not work for the early proposal versions of these utilities due to the way it specified
7322 handling of trailing slashes.

7323 Since the definition of *pathname* specifies implementation-defined behavior for pathnames
7324 starting with two slash characters, this volume of IEEE Std 1003.1-2001 specifies similar
7325 implementation-defined behavior for the *basename* and *dirname* utilities.

7326 **FUTURE DIRECTIONS**

7327 None.

7328 **SEE ALSO**

7329 Section 2.5 (on page 33), *dirname*

7330 **CHANGE HISTORY**

7331 First released in Issue 2.

7332 **Issue 6**

7333 IEEE PASC Interpretation 1003.2 #164 is applied.

7334 The normative text is reworded to avoid use of the term “must” for application requirements.

7335 **NAME**

7336 batch — schedule commands to be executed in a batch queue

7337 **SYNOPSIS**

7338 UP **batch**

7339

7340 **DESCRIPTION**

7341 The *batch* utility shall read commands from standard input and schedule them for execution in a
7342 batch queue. It shall be the equivalent of the command:

7343 **at -q b -m now**

7344 where queue *b* is a special *at* queue, specifically for batch jobs. Batch jobs shall be submitted to
7345 the batch queue with no time constraints and shall be run by the system using algorithms, based
7346 on unspecified factors, that may vary with each invocation of *batch*.

7347 XSI Users shall be permitted to use *batch* if their name appears in the file **/usr/lib/cron/at.allow**. If
7348 that file does not exist, the file **/usr/lib/cron/at.deny** shall be checked to determine whether the
7349 user shall be denied access to *batch*. If neither file exists, only a process with the appropriate
7350 privileges shall be allowed to submit a job. If only **at.deny** exists and is empty, global usage shall
7351 be permitted. The **at.allow** and **at.deny** files shall consist of one user name per line.

7352 **OPTIONS**

7353 None.

7354 **OPERANDS**

7355 None.

7356 **STDIN**

7357 The standard input shall be a text file consisting of commands acceptable to the shell command
7358 language described in Chapter 2 (on page 29).

7359 **INPUT FILES**

7360 XSI The text files **/usr/lib/cron/at.allow** and **/usr/lib/cron/at.deny** shall contain zero or more user
7361 names, one per line, of users who are, respectively, authorized or denied access to the *at* and
7362 *batch* utilities.

7363 **ENVIRONMENT VARIABLES**

7364 The following environment variables shall affect the execution of *batch*:

7365 **LANG** Provide a default value for the internationalization variables that are unset or null.
7366 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
7367 Internationalization Variables for the precedence of internationalization variables
7368 used to determine the values of locale categories.)

7369 **LC_ALL** If set to a non-empty string value, override the values of all the other
7370 internationalization variables.

7371 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
7372 characters (for example, single-byte as opposed to multi-byte characters in
7373 arguments and input files).

7374 **LC_MESSAGES**

7375 Determine the locale that should be used to affect the format and contents of
7376 diagnostic messages written to standard error and informative messages written to
7377 standard output.

7378 **LC_TIME** Determine the format and contents for date and time strings written by *batch*.

7379	XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
7380		SHELL	Determine the name of a command interpreter to be used to invoke the at-job. If the variable is unset or null, <i>sh</i> shall be used. If it is set to a value other than a name for <i>sh</i> , the implementation shall do one of the following: use that shell; use <i>sh</i> ; use the login shell from the user database; any of the preceding accompanied by a warning diagnostic about which was chosen.
7385		TZ	Determine the timezone. The job shall be submitted for execution at the time specified by <i>timespec</i> or -t time relative to the timezone specified by the TZ variable. If <i>timespec</i> specifies a timezone, it overrides TZ . If <i>timespec</i> does not specify a timezone and TZ is unset or null, an unspecified default timezone shall be used.
7390		ASYNCHRONOUS EVENTS	
7391			Default.
7392		STDOUT	When standard input is a terminal, prompts of unspecified format for each line of the user input described in the STDIN section may be written to standard output.
7395		STDERR	The following shall be written to standard error when a job has been successfully submitted: "job %s at %s\n", <i>at_job_id</i> , < <i>date</i> > where <i>date</i> shall be equivalent in format to the output of: <i>date</i> +"%a %b %e %T %Y" The date and time written shall be adjusted so that they appear in the timezone of the user (as determined by the TZ variable). Neither this, nor warning messages concerning the selection of the command interpreter, are considered a diagnostic that changes the exit status. Diagnostic messages, if any, shall be written to standard error.
7405		OUTPUT FILES	
7406			None.
7407		EXTENDED DESCRIPTION	
7408			None.
7409		EXIT STATUS	The following exit values shall be returned: 0 Successful completion. >0 An error occurred.
7413		CONSEQUENCES OF ERRORS	
7414			The job shall not be scheduled.

7415 **APPLICATION USAGE**

7416 It may be useful to redirect standard output within the specified commands.

7417 **EXAMPLES**

7418 1. This sequence can be used at a terminal:

```
7419     batch
7420         sort < file >outfile
7421         EOT
```

7422 2. This sequence, which demonstrates redirecting standard error to a pipe, is useful in a
7423 command procedure (the sequence of output redirection specifications is significant):

```
7424     batch <<
7425         ! diff file1 file2 2>&1 >outfile | mailx mygroup
7426         !
```

7427 **RATIONALE**

7428 Early proposals described *batch* in a manner totally separated from *at*, even though the historical
7429 model treated it almost as a synonym for *at -qb*. A number of features were added to list and
7430 control batch work separately from those in *at*. Upon further reflection, it was decided that the
7431 benefit of this did not merit the change to the historical interface.

7432 The **-m** option was included on the equivalent *at* command because it is historical practice to
7433 mail results to the submitter, even if all job-produced output is redirected. As explained in the
7434 RATIONALE for *at*, the **now** keyword submits the job for immediate execution (after scheduling
7435 delays), despite some historical systems where *at now* would have been considered an error.

7436 **FUTURE DIRECTIONS**

7437 None.

7438 **SEE ALSO**

7439 *at*

7440 **CHANGE HISTORY**

7441 First released in Issue 2.

7442 **Issue 6**

7443 This utility is marked as part of the User Portability Utilities option.

7444 The NAME is changed to align with the IEEE P1003.2b draft standard.

7445 The normative text is reworded to avoid use of the term “must” for application requirements.

7446 NAME

7447 bc — arbitrary-precision arithmetic language

7448 SYNOPSIS

7449 bc [-l] [*file* ...]

7450 DESCRIPTION

7451 The *bc* utility shall implement an arbitrary precision calculator. It shall take input from any files given, then read from the standard input. If the standard input and standard output to *bc* are attached to a terminal, the invocation of *bc* shall be considered to be *interactive*, causing behavioral constraints described in the following sections.

7455 OPTIONS

7456 The *bc* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

7458 The following option shall be supported:

7459 **-l** (The letter ell.) Define the math functions and initialize *scale* to 20, instead of the default zero; see the EXTENDED DESCRIPTION section.

7461 OPERANDS

7462 The following operand shall be supported:

7463 *file* A pathname of a text file containing *bc* program statements. After all *files* have been read, *bc* shall read the standard input.

7465 STDIN

7466 See the INPUT FILES section.

7467 INPUT FILES

7468 Input files shall be text files containing a sequence of comments, statements, and function definitions that shall be executed as they are read.

7470 ENVIRONMENT VARIABLES

7471 The following environment variables shall affect the execution of *bc*:

7472 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

7476 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

7478 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

7481 *LC_MESSAGES*

7482 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

7484 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

7485 ASYNCHRONOUS EVENTS

7486 Default.

7487 **STDOUT**

7488 The output of the *bc* utility shall be controlled by the program read, and consist of zero or more
 7489 lines containing the value of all executed expressions without assignments. The radix and
 7490 precision of the output shall be controlled by the values of the **obase** and **scale** variables; see the
 7491 EXTENDED DESCRIPTION section.

7492 **STDERR**

7493 The standard error shall be used only for diagnostic messages.

7494 **OUTPUT FILES**

7495 None.

7496 **EXTENDED DESCRIPTION**7497 **Grammar**

7498 The grammar in this section and the lexical conventions in the following section shall together
 7499 describe the syntax for *bc* programs. The general conventions for this style of grammar are
 7500 described in Section 1.10 (on page 19). A valid program can be represented as the non-terminal
 7501 symbol **program** in the grammar. This formal syntax shall take precedence over the text syntax
 7502 description.

```

7503     %token    EOF NEWLINE STRING LETTER NUMBER
7504     %token    MUL_OP
7505     /*      '*', '/', '%'           */
7506     %token    ASSIGN_OP
7507     /*      '=' , '+=' , '-=' , '*=' , '/=' , '%=' , '^=' */
7508     %token    REL_OP
7509     /*      '==' , '<=' , '>=' , '!=' , '<' , '>'        */
7510     %token    INCR_DECR
7511     /*      '++' , '--'           */
7512     %token    Define      Break      Quit      Length
7513     /*      'define' , 'break' , 'quit' , 'length'       */
7514     %token    Return      For       If        While      Sqrt
7515     /*      'return' , 'for' , 'if' , 'while' , 'sqrt'   */
7516     %token    Scale       Ibase     Obase     Auto
7517     /*      'scale' , 'ibase' , 'obase' , 'auto'      */
7518     %start   program
7519     %%
7520     program      : EOF
7521             | input_item program
7522             ;
7523     input_item   : semicolon_list NEWLINE
7524             | function
7525             ;
7526     semicolon_list : /* empty */
7527             | statement
7528             | semicolon_list ';' statement
7529             | semicolon_list ';'
```

```
7530 ;  
7531     statement_list : /* empty */  
7532     | statement  
7533     | statement_list NEWLINE  
7534     | statement_list NEWLINE statement  
7535     | statement_list ';' ;  
7536     | statement_list ';' statement  
7537 ;  
7538     statement : expression  
7539     | STRING  
7540     | Break  
7541     | Quit  
7542     | Return  
7543     | Return '(' return_expression ')' ;  
7544     | For '(' expression ';' ;  
7545         relational_expression ;  
7546         expression ')' statement  
7547     | If '(' relational_expression ')' statement  
7548     | While '(' relational_expression ')' statement  
7549     | '{' statement_list '}' ;  
7550 ;  
7551     function : Define LETTER '(' opt_parameter_list ')' ;  
7552         '{' NEWLINE opt_auto_define_list  
7553             statement_list '}' ;  
7554 ;  
7555     opt_parameter_list : /* empty */  
7556     | parameter_list  
7557 ;  
7558     parameter_list : LETTER  
7559     | define_list ',' LETTER  
7560 ;  
7561     opt_auto_define_list : /* empty */  
7562     | Auto define_list NEWLINE  
7563     | Auto define_list ';' ;  
7564 ;  
7565     define_list : LETTER  
7566     | LETTER '[' ']' ;  
7567     | define_list ',' LETTER  
7568     | define_list ',' LETTER '[' ']' ;  
7569 ;  
7570     opt_argument_list : /* empty */  
7571     | argument_list  
7572 ;  
7573     argument_list : expression  
7574     | LETTER '[' ']' ',' argument_list ;  
7575 ;
```

```

7576     relational_expression : expression
7577             | expression REL_OP expression
7578             ;
7579     return_expression   : /* empty */
7580             | expression
7581             ;
7582     expression        : named_expression
7583             | NUMBER
7584             | '(' expression ')'
7585             | LETTER '(' opt_argument_list ')'
7586             | '-' expression
7587             | expression '+' expression
7588             | expression '-' expression
7589             | expression MUL_OP expression
7590             | expression '^' expression
7591             | INCR_DECR named_expression
7592             | named_expression INCR_DECR
7593             | named_expression ASSIGN_OP expression
7594             | Length '(' expression ')'
7595             | Sqrt '(' expression ')'
7596             | Scale '(' expression ')'
7597             ;
7598     named_expression   : LETTER
7599             | LETTER '[' expression ']'
7600             | Scale
7601             | Ibase
7602             | Obase
7603             ;

```

7604 Lexical Conventions in bc

7605 The lexical conventions for *bc* programs, with respect to the preceding grammar, shall be as
7606 follows:

- 7607 1. Except as noted, *bc* shall recognize the longest possible token or delimiter beginning at a
7608 given point.
- 7609 2. A comment shall consist of any characters beginning with the two adjacent characters
7610 "/*" and terminated by the next occurrence of the two adjacent characters "*/".
7611 Comments shall have no effect except to delimit lexical tokens.
- 7612 3. The <newline> shall be recognized as the token NEWLINE.
- 7613 4. The token STRING shall represent a string constant; it shall consist of any characters
7614 beginning with the double-quote character ('') and terminated by another occurrence of
7615 the double-quote character. The value of the string is the sequence of all characters
7616 between, but not including, the two double-quote characters. All characters shall be taken
7617 literally from the input, and there is no way to specify a string containing a double-quote
7618 character. The length of the value of each string shall be limited to {BC_STRING_MAX}
7619 bytes.
- 7620 5. A <blank> shall have no effect except as an ordinary character if it appears within a
7621 STRING token, or to delimit a lexical token other than STRING.

- 7622 6. The combination of a backslash character immediately followed by a <newline> shall have
 7623 no effect other than to delimit lexical tokens with the following exceptions:

- 7624 • It shall be interpreted as the character sequence "\<newline>" in **STRING** tokens.
 7625 • It shall be ignored as part of a multi-line **NUMBER** token.

- 7626 7. The token **NUMBER** shall represent a numeric constant. It shall be recognized by the
 7627 following grammar:

```
7628        NUMBER : integer
  7629            | '.' integer
  7630            | integer '.'
  7631            | integer '.' integer
  7632            ;
  7633        integer : digit
  7634            | integer digit
  7635            ;
  7636        digit : 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
  7637            | 8 | 9 | A | B | C | D | E | F
  7638            ;
```

- 7639 8. The value of a **NUMBER** token shall be interpreted as a numeral in the base specified by
 7640 the value of the internal register **ibase** (described below). Each of the **digit** characters shall
 7641 have the value from 0 to 15 in the order listed here, and the period character shall represent
 7642 the radix point. The behavior is undefined if digits greater than or equal to the value of
 7643 **ibase** appear in the token. However, note the exception for single-digit values being
 7644 assigned to **ibase** and **obase** themselves, in **Operations in bc** (on page 198).

- 7645 9. The following keywords shall be recognized as tokens:

auto	ibase	length	return	while
break	if	obase	scale	
define	for	quit	sqrt	

- 7649 10. Any of the following characters occurring anywhere except within a keyword shall be
 7650 recognized as the token **LETTER**:

7651 a b c d e f g h i j k l m n o p q r s t u v w x y z

- 7652 11. The following single-character and two-character sequences shall be recognized as the
 7653 token **ASSIGN_OP**:

7654 = += -= *= /= %= ^=

- 7655 12. If an '=' character, as the beginning of a token, is followed by a '-' character with no
 7656 intervening delimiter, the behavior is undefined.

- 7657 13. The following single-characters shall be recognized as the token **MUL_OP**:

7658 * / %

- 7659 14. The following single-character and two-character sequences shall be recognized as the
 7660 token **REL_OP**:

7661 == <= >= != < >

- 7662 15. The following two-character sequences shall be recognized as the token **INCR_DECR**:

- 7663 ++ --
- 7664 16. The following single characters shall be recognized as tokens whose names are the
7665 character:
- 7666 $\langle\text{newline}\rangle$ $($ $)$ $,$ $+$ $-$ $;$ $[$ $]$ $^$ $\{$ $\}$
- 7667 17. The token **EOF** is returned when the end of input is reached.

7668 Operations in bc

7669 There are three kinds of identifiers: ordinary identifiers, array identifiers, and function
7670 identifiers. All three types consist of single lowercase letters. Array identifiers shall be followed
7671 by square brackets ("[]"). An array subscript is required except in an argument or auto list.
7672 Arrays are singly dimensioned and can contain up to {BC_DIM_MAX} elements. Indexing shall
7673 begin at zero so an array is indexed from 0 to {BC_DIM_MAX}-1. Subscripts shall be truncated
7674 to integers. The application shall ensure that function identifiers are followed by parentheses,
7675 possibly enclosing arguments. The three types of identifiers do not conflict.

7676 The following table summarizes the rules for precedence and associativity of all operators.
7677 Operators on the same line shall have the same precedence; rows are in order of decreasing
7678 precedence.

7679 **Table 4-3** Operators in *bc*

7680 Operator	7681 Associativity
7681 ++ , --	N/A
7682 unary $-$	N/A
7683 $^$	Right to left
7684 $*$, $/$, $\%$	Left to right
7685 $+$, binary $-$	Left to right
7686 $=$, $+=$, $-=$, $*=$, $/=$, $\%=$, $^=$	Right to left
7687 $==$, $<=$, $>=$, $!=$, $<$, $>$	None

7688 Each expression or named expression has a *scale*, which is the number of decimal digits that
7689 shall be maintained as the fractional portion of the expression.

7690 *Named expressions* are places where values are stored. Named expressions shall be valid on the
7691 left side of an assignment. The value of a named expression shall be the value stored in the place
7692 named. Simple identifiers and array elements are named expressions; they have an initial value
7693 of zero and an initial scale of zero.

7694 The internal registers **scale**, **ibase**, and **obase** are all named expressions. The scale of an
7695 expression consisting of the name of one of these registers shall be zero; values assigned to any
7696 of these registers are truncated to integers. The **scale** register shall contain a global value used in
7697 computing the scale of expressions (as described below). The value of the register **scale** is
7698 limited to $0 \leq \text{scale} \leq \{\text{BC_SCALE_MAX}\}$ and shall have a default value of zero. The **ibase** and
7699 **obase** registers are the input and output number radix, respectively. The value of **ibase** shall be
7700 limited to:

7701 $2 \leq \text{ibase} \leq 16$

7702 The value of **obase** shall be limited to:

7703 $2 \leq \text{obase} \leq \{\text{BC_BASE_MAX}\}$

7704 When either **ibase** or **obase** is assigned a single **digit** value from the list in **Lexical Conventions**
7705 in **bc** (on page 196), the value shall be assumed in hexadecimal. (For example, **ibase=A** sets to

7706 base ten, regardless of the current **ibase** value.) Otherwise, the behavior is undefined when
7707 digits greater than or equal to the value of **ibase** appear in the input. Both **ibase** and **obase** shall
7708 have initial values of 10.

7709 Internal computations shall be conducted as if in decimal, regardless of the input and output
7710 bases, to the specified number of decimal digits. When an exact result is not achieved (for
7711 example, **scale**=0; 3.2/1), the result shall be truncated.

7712 For all values of **obase** specified by this volume of IEEE Std 1003.1-2001, *bc* shall output numeric
7713 values by performing each of the following steps in order:

- 7714 1. If the value is less than zero, a hyphen ('-') character shall be output.
- 7715 2. One of the following is output, depending on the numerical value:
 - 7716 • If the absolute value of the numerical value is greater than or equal to one, the integer
7717 portion of the value shall be output as a series of digits appropriate to **obase** (as
7718 described below), most significant digit first. The most significant non-zero digit shall
7719 be output next, followed by each successively less significant digit.
 - 7720 • If the absolute value of the numerical value is less than one but greater than zero and
7721 the scale of the numerical value is greater than zero, it is unspecified whether the
7722 character 0 is output.
 - 7723 • If the numerical value is zero, the character 0 shall be output.
- 7724 3. If the scale of the value is greater than zero and the numeric value is not zero, a period
7725 character shall be output, followed by a series of digits appropriate to **obase** (as described
7726 below) representing the most significant portion of the fractional part of the value. If *s*
7727 represents the scale of the value being output, the number of digits output shall be *s* if
7728 **obase** is 10, less than or equal to *s* if **obase** is greater than 10, or greater than or equal to *s* if
7729 **obase** is less than 10. For **obase** values other than 10, this should be the number of digits
7730 needed to represent a precision of 10^{-s} .

7731 For **obase** values from 2 to 16, valid digits are the first **obase** of the single characters:

7732 0 1 2 3 4 5 6 7 8 9 A B C D E F

7733 which represent the values zero to 15, inclusive, respectively.

7734 For bases greater than 16, each digit shall be written as a separate multi-digit decimal number.
7735 Each digit except the most significant fractional digit shall be preceded by a single <space>. For
7736 bases from 17 to 100, *bc* shall write two-digit decimal numbers; for bases from 101 to 1000,
7737 three-digit decimal strings, and so on. For example, the decimal number 1 024 in base 25 would
7738 be written as:

7739 Δ01Δ15Δ24

7740 and in base 125, as:

7741 Δ008Δ024

7742 Very large numbers shall be split across lines with 70 characters per line in the POSIX locale;
7743 other locales may split at different character boundaries. Lines that are continued shall end with
7744 a backslash ('\').

7745 A function call shall consist of a function name followed by parentheses containing a comma-
7746 separated list of expressions, which are the function arguments. A whole array passed as an
7747 argument shall be specified by the array name followed by empty square brackets. All function
7748 arguments shall be passed by value. As a result, changes made to the formal parameters shall
7749 have no effect on the actual arguments. If the function terminates by executing a **return**

7750 statement, the value of the function shall be the value of the expression in the parentheses of the
7751 **return** statement or shall be zero if no expression is provided or if there is no **return** statement.

7752 The result of **sqrt(expression)** shall be the square root of the expression. The result shall be
7753 truncated in the least significant decimal place. The scale of the result shall be the scale of the
7754 expression or the value of **scale**, whichever is larger.

7755 The result of **length(expression)** shall be the total number of significant decimal digits in the
7756 expression. The scale of the result shall be zero.

7757 The result of **scale(expression)** shall be the scale of the expression. The scale of the result shall be
7758 zero.

7759 A numeric constant shall be an expression. The scale shall be the number of digits that follow the
7760 radix point in the input representing the constant, or zero if no radix point appears.

7761 The sequence (*expression*) shall be an expression with the same value and scale as *expression*.
7762 The parentheses can be used to alter the normal precedence.

7763 The semantics of the unary and binary operators are as follows:

7764 **-expression**

7765 The result shall be the negative of the *expression*. The scale of the result shall be the scale of
7766 *expression*.

7767 The unary increment and decrement operators shall not modify the scale of the named
7768 expression upon which they operate. The scale of the result shall be the scale of that named
7769 expression.

7770 **++named-expression**

7771 The named expression shall be incremented by one. The result shall be the value of the
7772 named expression after incrementing.

7773 **--named-expression**

7774 The named expression shall be decremented by one. The result shall be the value of the
7775 named expression after decrementing.

7776 **named-expression++**

7777 The named expression shall be incremented by one. The result shall be the value of the
7778 named expression before incrementing.

7779 **named-expression--**

7780 The named expression shall be decremented by one. The result shall be the value of the
7781 named expression before decrementing.

7782 The exponentiation operator, circumflex ('^'), shall bind right to left.

7783 **expression^expression**

7784 The result shall be the first *expression* raised to the power of the second *expression*. If the
7785 second expression is not an integer, the behavior is undefined. If *a* is the scale of the left
7786 expression and *b* is the absolute value of the right expression, the scale of the result shall be:

7787 **if b >= 0 min(a * b, max(scale, a)) if b < 0 scale**

7788 The multiplicative operators ('*', '/', '%') shall bind left to right.

7789 **expression*expression**

7790 The result shall be the product of the two expressions. If *a* and *b* are the scales of the two
7791 expressions, then the scale of the result shall be:

```

7792      min(a+b,max(scale,a,b))
7793  expression/expression
7794      The result shall be the quotient of the two expressions. The scale of the result shall be the
7795      value of scale.
7796  expression%expression
7797      For expressions a and b, a%b shall be evaluated equivalent to the steps:
7798          1. Compute a/b to current scale.
7799          2. Use the result to compute:
7800              a - (a / b) * b
7801              to scale:
7802              max(scale + scale(b), scale(a))
7803      The scale of the result shall be:
7804      max(scale + scale(b), scale(a))
7805      When scale is zero, the '%' operator is the mathematical remainder operator.
7806      The additive operators ('+', '-') shall bind left to right.
7807  expression+expression
7808      The result shall be the sum of the two expressions. The scale of the result shall be the
7809      maximum of the scales of the expressions.
7810  expression-expression
7811      The result shall be the difference of the two expressions. The scale of the result shall be the
7812      maximum of the scales of the expressions.
7813      The assignment operators ('=' , "+=" , "-=" , "*=" , "/=" , "%=" , "^=") shall bind right to left.
7814  named-expression=expression
7815      This expression shall result in assigning the value of the expression on the right to the
7816      named expression on the left. The scale of both the named expression and the result shall be
7817      the scale of expression.
7818      The compound assignment forms:
7819      named-expression <operator>= expression
7820      shall be equivalent to:
7821      named-expression=named-expression <operator> expression
7822      except that the named-expression shall be evaluated only once.
7823      Unlike all other operators, the relational operators ('<', '>', "<=", ">=", "==" , "!=")
7824      shall be only valid as the object of an if, while, or inside a for statement.
7825  expression1<expression2
7826      The relation shall be true if the value of expression1 is strictly less than the value of
7827      expression2.
7828  expression1>expression2
7829      The relation shall be true if the value of expression1 is strictly greater than the value of
7830      expression2.

```

7831 *expression1<=expression2*

7832 The relation shall be true if the value of *expression1* is less than or equal to the value of
7833 *expression2*.

7834 *expression1>=expression2*

7835 The relation shall be true if the value of *expression1* is greater than or equal to the value of
7836 *expression2*.

7837 *expression1==expression2*

7838 The relation shall be true if the values of *expression1* and *expression2* are equal.

7839 *expression1!=expression2*

7840 The relation shall be true if the values of *expression1* and *expression2* are unequal.

7841 There are only two storage classes in *bc*: global and automatic (local). Only identifiers that are
7842 local to a function need be declared with the **auto** command. The arguments to a function shall
7843 be local to the function. All other identifiers are assumed to be global and available to all
7844 functions. All identifiers, global and local, have initial values of zero. Identifiers declared as auto
7845 shall be allocated on entry to the function and released on returning from the function. They
7846 therefore do not retain values between function calls. Auto arrays shall be specified by the array
7847 name followed by empty square brackets. On entry to a function, the old values of the names
7848 that appear as parameters and as automatic variables shall be pushed onto a stack. Until the
7849 function returns, reference to these names shall refer only to the new values.

7850 References to any of these names from other functions that are called from this function also
7851 refer to the new value until one of those functions uses the same name for a local variable.

7852 When a statement is an expression, unless the main operator is an assignment, execution of the
7853 statement shall write the value of the expression followed by a <newline>.

7854 When a statement is a string, execution of the statement shall write the value of the string.

7855 Statements separated by semicolons or <newline>s shall be executed sequentially. In an
7856 interactive invocation of *bc*, each time a <newline> is read that satisfies the grammatical
7857 production:

7858 *input_item* : *semicolon_list* NEWLINE

7859 the sequential list of statements making up the **semicolon_list** shall be executed immediately
7860 and any output produced by that execution shall be written without any delay due to buffering.

7861 In an **if** statement (**if**(*relation*) *statement*), the *statement* shall be executed if the *relation* is true.

7862 The **while** statement (**while**(*relation*) *statement*) implements a loop in which the *relation* is tested;
7863 each time the *relation* is true, the *statement* shall be executed and the *relation* retested. When the
7864 *relation* is false, execution shall resume after *statement*.

7865 A **for** statement(**for**(*expression*; *relation*; *expression*) *statement*) shall be the same as:

7866 *first-expression*
7867 **while** (*relation*) {
7868 *statement*
7869 *last-expression*
7870 }

7871 The application shall ensure that all three expressions are present.

7872 The **break** statement shall cause termination of a **for** or **while** statement.

7873 The **auto** statement (**auto** *identifier* [,*identifier*] ...) shall cause the values of the *identifiers* to be
7874 pushed down. The *identifiers* can be ordinary *identifiers* or *array identifiers*. *Array identifiers*

7875 shall be specified by following the array name by empty square brackets. The application shall
 7876 ensure that the **auto** statement is the first statement in a function definition.

7877 A **define** statement:

```
7878 define LETTER ( opt_parameter_list ) {
7879     opt_auto_define_list
7880     statement_list
7881 }
```

7882 defines a function named **LETTER**. If a function named **LETTER** was previously defined, the
 7883 **define** statement shall replace the previous definition. The expression:

```
7884 LETTER ( opt_argument_list )
```

7885 shall invoke the function named **LETTER**. The behavior is undefined if the number of
 7886 arguments in the invocation does not match the number of parameters in the definition.
 7887 Functions shall be defined before they are invoked. A function shall be considered to be defined
 7888 within its own body, so recursive calls are valid. The values of numeric constants within a
 7889 function shall be interpreted in the base specified by the value of the **ibase** register when the
 7890 function is invoked.

7891 The **return** statements (**return** and **return(expression)**) shall cause termination of a function,
 7892 popping of its auto variables, and specification of the result of the function. The first form shall
 7893 be equivalent to **return(0)**. The value and scale of the result returned by the function shall be the
 7894 value and scale of the expression returned.

7895 The **quit** statement (**quit**) shall stop execution of a *bc* program at the point where the statement
 7896 occurs in the input, even if it occurs in a function definition, or in an **if**, **for**, or **while** statement.

7897 The following functions shall be defined when the **-l** option is specified:

7898 **s(expression)**
 7899 Sine of argument in radians.

7900 **c(expression)**
 7901 Cosine of argument in radians.

7902 **a(expression)**
 7903 Arctangent of argument.

7904 **l(expression)**
 7905 Natural logarithm of argument.

7906 **e(expression)**
 7907 Exponential function of argument.

7908 **j(expression, expression)**
 7909 Bessel function of integer order.

7910 The scale of the result returned by these functions shall be the value of the **scale** register at the
 7911 time the function is invoked. The value of the **scale** register after these functions have completed
 7912 their execution shall be the same value it had upon invocation. The behavior is undefined if any
 7913 of these functions is invoked with an argument outside the domain of the mathematical
 7914 function.

7915 **EXIT STATUS**

7916 The following exit values shall be returned:

7917 0 All input files were processed successfully.

7918 *unspecified* An error occurred.

7919 CONSEQUENCES OF ERRORS

7920 If any *file* operand is specified and the named file cannot be accessed, *bc* shall write a diagnostic
7921 message to standard error and terminate without any further action.

7922 In an interactive invocation of *bc*, the utility should print an error message and recover following
7923 any error in the input. In a non-interactive invocation of *bc*, invalid input causes undefined
7924 behavior.

7925 APPLICATION USAGE

7926 Automatic variables in *bc* do not work in exactly the same way as in either C or PL/1.

7927 For historical reasons, the exit status from *bc* cannot be relied upon to indicate that an error has
7928 occurred. Returning zero after an error is possible. Therefore, *bc* should be used primarily by
7929 interactive users (who can react to error messages) or by application programs that can
7930 somehow validate the answers returned as not including error messages.

7931 The *bc* utility always uses the period ('.') character to represent a radix point, regardless of any
7932 decimal-point character specified as part of the current locale. In languages like C or awk, the
7933 period character is used in program source, so it can be portable and unambiguous, while the
7934 locale-specific character is used in input and output. Because there is no distinction between
7935 source and input in *bc*, this arrangement would not be possible. Using the locale-specific
7936 character in *bc*'s input would introduce ambiguities into the language; consider the following
7937 example in a locale with a comma as the decimal-point character:

```
7938 define f(a,b) {  
7939     ...  
7940 }  
7941 ...  
7942 f(1,2,3)
```

7943 Because of such ambiguities, the period character is used in input. Having input follow different
7944 conventions from output would be confusing in either pipeline usage or interactive usage, so the
7945 period is also used in output.

7946 EXAMPLES

7947 In the shell, the following assigns an approximation of the first ten digits of ' π ' to the variable
7948 *x*:

```
7949 x=$(printf "%s\n" 'scale = 10; 104348/33215' | bc)
```

7950 The following *bc* program prints the same approximation of ' π ', with a label, to standard
7951 output:

```
7952 scale = 10  
7953 "pi equals "  
7954 104348 / 33215
```

7955 The following defines a function to compute an approximate value of the exponential function
7956 (note that such a function is predefined if the **-l** option is specified):

```
7957 scale = 20  
7958 define e(x){  
7959     auto a, b, c, i, s  
7960     a = 1  
7961     b = 1  
7962     s = 1
```

```

7963     for (i = 1; i == 1; i++) {
7964         a = a*x
7965         b = b*i
7966         c = a/b
7967         if (c == 0) {
7968             return(s)
7969         }
7970         s = s+c
7971     }
7972 }
```

7973 The following prints approximate values of the exponential function of the first ten integers:

```

7974     for (i = 1; i <= 10; ++i) {
7975         e(i)
7976     }
```

7977 RATIONALE

7978 The *bc* utility is implemented historically as a front-end processor for *dc*; *dc* was not selected to
 7979 be part of this volume of IEEE Std 1003.1-2001 because *bc* was thought to have a more intuitive
 7980 programmatic interface. Current implementations that implement *bc* using *dc* are expected to be
 7981 compliant.

7982 The exit status for error conditions has been left unspecified for several reasons:

- 7983 • The *bc* utility is used in both interactive and non-interactive situations. Different exit codes
 7984 may be appropriate for the two uses.
- 7985 • It is unclear when a non-zero exit should be given; divide-by-zero, undefined functions, and
 7986 syntax errors are all possibilities.
- 7987 • It is not clear what utility the exit status has.
- 7988 • In the 4.3 BSD, System V, and Ninth Edition implementations, *bc* works in conjunction with
 7989 *dc*. The *dc* utility is the parent, *bc* is the child. This was done to cleanly terminate *bc* if *dc*
 7990 aborted.

7991 The decision to have *bc* exit upon encountering an inaccessible input file is based on the belief
 7992 that *bc file1 file2* is used most often when at least *file1* contains data/function
 7993 declarations/initializations. Having *bc* continue with prerequisite files missing is probably not
 7994 useful. There is no implication in the CONSEQUENCES OF ERRORS section that *bc* must check
 7995 all its files for accessibility before opening any of them.

7996 There was considerable debate on the appropriateness of the language accepted by *bc*. Several
 7997 reviewers preferred to see either a pure subset of the C language or some changes to make the
 7998 language more compatible with C. While the *bc* language has some obvious similarities to C, it
 7999 has never claimed to be compatible with any version of C. An interpreter for a subset of C might
 8000 be a very worthwhile utility, and it could potentially make *bc* obsolete. However, no such utility
 8001 is known in historical practice, and it was not within the scope of this volume of
 8002 IEEE Std 1003.1-2001 to define such a language and utility. If and when they are defined, it may
 8003 be appropriate to include them in a future version of IEEE Std 1003.1. This left the following
 8004 alternatives:

- 8005 1. Exclude any calculator language from this volume of IEEE Std 1003.1-2001.

8006 The consensus of the standard developers was that a simple programmatic calculator
 8007 language is very useful for both applications and interactive users. The only arguments for
 8008 excluding any calculator were that it would become obsolete if and when a C-compatible

8009 one emerged, or that the absence would encourage the development of such a C-
8010 compatible one. These arguments did not sufficiently address the needs of current
8011 application writers.

8012 2. Standardize the historical *dc*, possibly with minor modifications.

8013 The consensus of the standard developers was that *dc* is a fundamentally less usable
8014 language and that that would be far too severe a penalty for avoiding the issue of being
8015 similar to but incompatible with C.

8016 3. Standardize the historical *bc*, possibly with minor modifications.

8017 This was the approach taken. Most of the proponents of changing the language would not
8018 have been satisfied until most or all of the incompatibilities with C were resolved. Since
8019 most of the changes considered most desirable would break historical applications and
8020 require significant modification to historical implementations, almost no modifications
8021 were made. The one significant modification that was made was the replacement of the
8022 historical *bc* assignment operators " $=+$ ", and so on, with the more modern " $+=$ ", and so
8023 on. The older versions are considered to be fundamentally flawed because of the lexical
8024 ambiguity in uses like $a=-1$.

8025 In order to permit implementations to deal with backwards-compatibility as they see fit,
8026 the behavior of this one ambiguous construct was made undefined. (At least three
8027 implementations have been known to support this change already, so the degree of change
8028 involved should not be great.)

8029 The '%' operator is the mathematical remainder operator when **scale** is zero. The behavior of
8030 this operator for other values of **scale** is from historical implementations of *bc*, and has been
8031 maintained for the sake of historical applications despite its non-intuitive nature.

8032 Historical implementations permit setting **ibase** and **obase** to a broader range of values. This
8033 includes values less than 2, which were not seen as sufficiently useful to standardize. These
8034 implementations do not interpret input properly for values of **ibase** that are greater than 16. This
8035 is because numeric constants are recognized syntactically, rather than lexically, as described in
8036 this volume of IEEE Std 1003.1-2001. They are built from lexical tokens of single hexadecimal
8037 digits and periods. Since <blank>s between tokens are not visible at the syntactic level, it is not
8038 possible to recognize the multi-digit "digits" used in the higher bases properly. The ability to
8039 recognize input in these bases was not considered useful enough to require modifying these
8040 implementations. Note that the recognition of numeric constants at the syntactic level is not a
8041 problem with conformance to this volume of IEEE Std 1003.1-2001, as it does not impact the
8042 behavior of conforming applications (and correct *bc* programs). Historical implementations also
8043 accept input with all of the digits '0'-'9' and 'A'-'F' regardless of the value of **ibase**; since
8044 digits with value greater than or equal to **ibase** are not really appropriate, the behavior when
8045 they appear is undefined, except for the common case of:

```
8046 ibase=8;
8047     /* Process in octal base. */
8048 ...
8049 ibase=A
8050     /* Restore decimal base. */
```

8051 In some historical implementations, if the expression to be written is an uninitialized array
8052 element, a leading <space> and/or up to four leading 0 characters may be output before the
8053 character zero. This behavior is considered a bug; it is unlikely that any currently conforming
8054 application relies on:

8055 echo 'b[3]' | bc
8056 returning 00000 rather than 0.

8057 Exact calculation of the number of fractional digits to output for a given value in a base other
8058 than 10 can be computationally expensive. Historical implementations use a faster
8059 approximation, and this is permitted. Note that the requirements apply only to values of **obase**
8060 that this volume of IEEE Std 1003.1-2001 requires implementations to support (in particular, not
8061 to 1, 0, or negative bases, if an implementation supports them as an extension).

8062 Historical implementations of *bc* did not allow array parameters to be passed as the last
8063 parameter to a function. New implementations are encouraged to remove this restriction even
8064 though it is not required by the grammar.

8065 **FUTURE DIRECTIONS**
8066 None.

8067 **SEE ALSO**
8068 Section 1.10 (on page 19), *awk*

8069 **CHANGE HISTORY**
8070 First released in Issue 4.

8071 **Issue 5**
8072 The FUTURE DIRECTIONS section is added.

8073 **Issue 6**
8074 Updated to align with the IEEE P1003.2b draft standard, which included resolution of several
8075 interpretations of the ISO POSIX-2:1993 standard.

8076 The normative text is reworded to avoid use of the term “must” for application requirements.

8077 **NAME**

8078 *bg* — run jobs in the background

8079 **SYNOPSIS**

8080 UP *bg* [*job_id* . . .]

8081

8082 **DESCRIPTION**

8083 If job control is enabled (see the description of *set -m*), the *bg* utility shall resume suspended jobs
8084 from the current environment (see Section 2.12 (on page 61)) by running them as background
8085 jobs. If the job specified by *job_id* is already a running background job, the *bg* utility shall have no
8086 effect and shall exit successfully.

8087 Using *bg* to place a job into the background shall cause its process ID to become “known in the
8088 current shell execution environment”, as if it had been started as an asynchronous list; see
8089 Section 2.9.3.1 (on page 50).

8090 **OPTIONS**

8091 None.

8092 **OPERANDS**

8093 The following operand shall be supported:

8094 *job_id* Specify the job to be resumed as a background job. If no *job_id* operand is given,
8095 the most recently suspended job shall be used. The format of *job_id* is described in
8096 the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.203, Job Control Job
8097 ID.

8098 **STDIN**

8099 Not used.

8100 **INPUT FILES**

8101 None.

8102 **ENVIRONMENT VARIABLES**

8103 The following environment variables shall affect the execution of *bg*:

8104 *LANG* Provide a default value for the internationalization variables that are unset or null.
8105 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
8106 Internationalization Variables for the precedence of internationalization variables
8107 used to determine the values of locale categories.)

8108 *LC_ALL* If set to a non-empty string value, override the values of all the other
8109 internationalization variables.

8110 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
8111 characters (for example, single-byte as opposed to multi-byte characters in
8112 arguments).

8113 ***LC_MESSAGES***

8114 Determine the locale that should be used to affect the format and contents of
8115 diagnostic messages written to standard error.

8116 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

8117 **ASYNCHRONOUS EVENTS**

8118 Default.

8119 STDOUT

8120 The output of *bg* shall consist of a line in the format:

8121 " [%d] %s\n", <*job-number*>, <*command*>

8122 where the fields are as follows:

8123 <*job-number*> A number that can be used to identify the job to the *wait*, *fg*, and *kill* utilities. Using
8124 these utilities, the job can be identified by prefixing the job number with '%'.

8125 <*command*> The associated command that was given to the shell.

8126 STDERR

8127 The standard error shall be used only for diagnostic messages.

8128 OUTPUT FILES

8129 None.

8130 EXTENDED DESCRIPTION

8131 None.

8132 EXIT STATUS

8133 The following exit values shall be returned:

8134 0 Successful completion.

8135 >0 An error occurred.

8136 CONSEQUENCES OF ERRORS

8137 If job control is disabled, the *bg* utility shall exit with an error and no job shall be placed in the
8138 background.

8139 APPLICATION USAGE

8140 A job is generally suspended by typing the SUSP character (<control>-Z on most systems); see
8141 the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. At
8142 that point, *bg* can put the job into the background. This is most effective when the job is
8143 expecting no terminal input and its output has been redirected to non-terminal files. A
8144 background job can be forced to stop when it has terminal output by issuing the command:

8145 stty tostop

8146 A background job can be stopped with the command:

8147 kill -s stop *job ID*

8148 The *bg* utility does not work as expected when it is operating in its own utility execution
8149 environment because that environment has no suspended jobs. In the following examples:

8150 ... | xargs bg
8151 (bg)

8152 each *bg* operates in a different environment and does not share its parent shell's understanding
8153 of jobs. For this reason, *bg* is generally implemented as a shell regular built-in.

8154 EXAMPLES

8155 None.

8156 RATIONALE

8157 The extensions to the shell specified in this volume of IEEE Std 1003.1-2001 have mostly been
8158 based on features provided by the KornShell. The job control features provided by *bg*, *fg*, and *jobs*
8159 are also based on the KornShell. The standard developers examined the characteristics of the C
8160 shell versions of these utilities and found that differences exist. Despite widespread use of the C

8161 shell, the KornShell versions were selected for this volume of IEEE Std 1003.1-2001 to maintain a
8162 degree of uniformity with the rest of the KornShell features selected (such as the very popular
8163 command line editing features).

8164 The *bg* utility is expected to wrap its output if the output exceeds the number of display
8165 columns.

8166 **FUTURE DIRECTIONS**

8167 None.

8168 **SEE ALSO**

8169 Section 2.9.3.1 (on page 50), *fg*, *kill*, *jobs*, *wait*

8170 **CHANGE HISTORY**

8171 First released in Issue 4.

8172 **Issue 6**

8173 This utility is marked as part of the User Portability Utilities option.

8174 The JC margin marker on the SYNOPSIS is removed since support for Job Control is mandatory
8175 in this issue. This is a FIPS requirement.

8176 **NAME**

8177 c99 — compile standard C programs

8178 **SYNOPSIS**8179 CD c99 [-c][-D name[=value]]...[-E][-g][-I directory] ... [-L directory]
8180 ... [-o outfile][-Oopt level][-s][-U name]... operand ...
81818182 **DESCRIPTION**

8183 The *c99* utility is an interface to the standard C compilation system; it shall accept source code
8184 conforming to the ISO C standard. The system conceptually consists of a compiler and link
8185 editor. The files referenced by *operands* shall be compiled and linked to produce an executable
8186 file. (It is unspecified whether the linking occurs entirely within the operation of *c99*; some
8187 implementations may produce objects that are not fully resolved until the file is executed.)

8188 If the **-c** option is specified, for all pathname operands of the form *file.c*, the files:

8189 \$(basename *pathname* .c).o

8190 shall be created as the result of successful compilation. If the **-c** option is not specified, it is
8191 unspecified whether such **.o** files are created or deleted for the *file.c* operands.

8192 If there are no options that prevent link editing (such as **-c** or **-E**), and all operands compile and
8193 link without error, the resulting executable file shall be written according to the **-o** *outfile* option
8194 (if present) or to the file **a.out**.

8195 The executable file shall be created as specified in Section 1.7.1.4 (on page 4), except that the file
8196 permission bits shall be set to:

8197 S_IRWXO | S_IRWXG | S_IROWXU

8198 and the bits specified by the *umask* of the process shall be cleared.

8199 **OPTIONS**

8200 The *c99* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
8201 12.2, Utility Syntax Guidelines, except that:

- The **-I** *library* operands have the format of options, but their position within a list of
8203 operands affects the order in which libraries are searched.
- The order of specifying the **-I** and **-L** options is significant.
- Conforming applications shall specify each option separately; that is, grouping option letters
8206 (for example, **-cO**) need not be recognized by all implementations.

8207 The following options shall be supported:

- c** Suppress the link-edit phase of the compilation, and do not remove any object files
8209 that are produced.
- g** Produce symbolic information in the object or executable files; the nature of this
8211 information is unspecified, and may be modified by implementation-defined
8212 interactions with other options.
- s** Produce object or executable files, or both, from which symbolic and other
8214 information not required for proper execution using the *exec* family defined in the
8215 System Interfaces volume of IEEE Std 1003.1-2001 has been removed (stripped). If
8216 both **-g** and **-s** options are present, the action taken is unspecified.
- o** *outfile* Use the pathname *outfile*, instead of the default **a.out**, for the executable file
8218 produced. If the **-o** option is present with **-c** or **-E**, the result is unspecified.

- 8219 **-D *name[=value]***
 8220 Define *name* as if by a C-language **#define** directive. If no *=value* is given, a value of
 8221 1 shall be used. The **-D** option has lower precedence than the **-U** option. That is, if
 8222 *name* is used in both a **-U** and a **-D** option, *name* shall be undefined regardless of
 8223 the order of the options. Additional implementation-defined *names* may be
 8224 provided by the compiler. Implementations shall support at least 2 048 bytes of **-D**
 8225 definitions and 256 *names*.
- 8226 **-E**
 8227 Copy C-language source files to standard output, expanding all preprocessor
 8228 directives; no compilation shall be performed. If any operand is not a text file, the
 effects are unspecified.
- 8229 **-I *directory***
 8230 Change the algorithm for searching for headers whose names are not absolute
 8231 pathnames to look in the directory named by the *directory* pathname before
 8232 looking in the usual places. Thus, headers whose names are enclosed in double-
 8233 quotes (" ") shall be searched for first in the directory of the file with the **#include**
 8234 line, then in directories named in **-I** options, and last in the usual places. For
 8235 headers whose names are enclosed in angle brackets ("<>"), the header shall be
 8236 searched for only in directories named in **-I** options and then in the usual places.
 8237 Directories named in **-I** options shall be searched in the order specified.
 8238 Implementations shall support at least ten instances of this option in a single *c99*
 command invocation.
- 8239 **-L *directory***
 8240 Change the algorithm of searching for the libraries named in the **-l** objects to look
 8241 in the directory named by the *directory* pathname before looking in the usual
 8242 places. Directories named in **-L** options shall be searched in the order specified.
 8243 Implementations shall support at least ten instances of this option in a single *c99*
 8244 command invocation. If a directory specified by a **-L** option contains files named
libc.a, **libm.a**, **libl.a**, or **liby.a**, the results are unspecified.
- 8245 **-O *optlevel***
 8246 Specify the level of code optimization. If the *optlevel* option-argument is the digit
 8247 '0', all special code optimizations shall be disabled. If it is the digit '1', the
 8248 nature of the optimization is unspecified. If the **-O** option is omitted, the nature of
 8249 the system's default optimization is unspecified. It is unspecified whether code
 8250 generated in the presence of the **-O 0** option is the same as that generated when
-O is omitted. Other *optlevel* values may be supported.
- 8251 **-U *name***
 8252 Remove any initial definition of *name*.
 Multiple instances of the **-D**, **-I**, **-U**, and **-L** options can be specified.

8253 OPERANDS

8254 An *operand* is either in the form of a pathname or the form **-l *library***. The application shall
 8255 ensure that at least one operand of the pathname form is specified. The following operands shall
 8256 be supported:

- 8257 **file.c**
 8258 A C-language source file to be compiled and optionally linked. The application
 shall ensure that the operand is of this form if the **-c** option is used.
- 8259 **file.a**
 8260 A library of object files typically produced by the *ar* utility, and passed directly to
 8261 the link editor. Implementations may recognize implementation-defined suffixes
 other than **.a** as denoting object file libraries.
- 8262 **file.o**
 8263 An object file produced by *c99 -c* and passed directly to the link editor.
 8264 Implementations may recognize implementation-defined suffixes other than **.o** as
 denoting object files.

8265 The processing of other files is implementation-defined.

8266 ***-l library*** (The letter ell.) Search the library named:
 liblibrary.a

8268 A library shall be searched when its name is encountered, so the placement of a ***-l*** operand is significant. Several standard libraries can be specified in this manner, as described in the EXTENDED DESCRIPTION section. Implementations may recognize implementation-defined suffixes other than **.a** as denoting libraries.

8272 **STDIN**

8273 Not used.

8274 **INPUT FILES**

8275 The input file shall be one of the following: a text file containing a C-language source program, an object file in the format produced by *c99 -c*, or a library of object files, in the format produced by archiving zero or more object files, using *ar*. Implementations may supply additional utilities that produce files in these formats. Additional input file formats are implementation-defined.

8279 **ENVIRONMENT VARIABLES**

8280 The following environment variables shall affect the execution of *c99*:

8281	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
8285	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
8287	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).
8290	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
8293 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
8294 XSI	<i>TMPDIR</i>	Provide a pathname that should override the default directory for temporary files, if any. On XSI-conforming systems, provide a pathname that shall override the default directory for temporary files, if any.

8297 **ASYNCHRONOUS EVENTS**

8298 Default.

8299 **STDOUT**

8300 If more than one *file* operand ending in **.c** (or possibly other unspecified suffixes) is given, for each such file:

8302 **"%s:\n"**, *<file>*

8303 may be written. These messages, if written, shall precede the processing of each input file; they shall not be written to the standard output if they are written to the standard error, as described in the STDERR section.

8306 If the ***-E*** option is specified, the standard output shall be a text file that represents the results of the preprocessing stage of the language; it may contain extra information appropriate for subsequent compilation passes.

8309 **STDERR**

8310 The standard error shall be used only for diagnostic messages. If more than one *file* operand
 8311 ending in .c (or possibly other unspecified suffixes) is given, for each such file:

8312 "%s:\n", <file>

8313 may be written to allow identification of the diagnostic and warning messages with the
 8314 appropriate input file. These messages, if written, shall precede the processing of each input file;
 8315 they shall not be written to the standard error if they are written to the standard output, as
 8316 described in the STDOUT section.

8317 This utility may produce warning messages about certain conditions that do not warrant
 8318 returning an error (non-zero) exit value.

8319 **OUTPUT FILES**

8320 Object files or executable files or both are produced in unspecified formats.

8321 **EXTENDED DESCRIPTION**8322 **Standard Libraries**

8323 The *c99* utility shall recognize the following -l operands for standard libraries:

- 8324 **-l c** This operand shall make visible all functions referenced in the System Interfaces
 volume of IEEE Std 1003.1-2001, with the possible exception of those functions
 listed as residing in <aio.h>, <arpa/inet.h>, <math.h>, <mqueue.h>, <netdb.h>,
 <netinet/in.h>, <pthread.h>, <sched.h>, <semaphore.h>, <spawn.h>,
 <sys/socket.h>, pthread_kill(), pthread_sigmask() in <signal.h>, <trace.h>,
 functions marked as extensions other than as part of the MF or MPR extensions in
 <sys/mman.h>, functions marked as ADV in <fcntl.h>, and functions marked as
 CS, CPT, and TMR in <time.h>. This operand shall not be required to be present to
 cause a search of this library.
- 8333 **-l l** This operand shall make visible all functions required by the C-language output of
 lex that are not made available through the -l c operand.
- 8335 **-l pthread** This operand shall make visible all functions referenced in <pthread.h> and
 pthread_kill() and pthread_sigmask() referenced in <signal.h>. An implementation
 may search this library in the absence of this operand.
- 8338 **-l m** This operand shall make visible all functions referenced in <math.h>. An
 implementation may search this library in the absence of this operand.
- 8340 **-l rt** This operand shall make visible all functions referenced in <aio.h>, <mqueue.h>,
 <sched.h>, <semaphore.h>, and <spawn.h>, functions marked as extensions other
 than as part of the MF or MPR extensions in <sys/mman.h>, functions marked as
 ADV in <fcntl.h>, and functions marked as CS, CPT, and TMR in <time.h>. An
 implementation may search this library in the absence of this operand.
- 8345 **-l trace** This operand shall make visible all functions referenced in <trace.h>. An
 implementation may search this library in the absence of this operand.
- 8347 **-l xnet** This operand makes visible all functions referenced in <arpa/inet.h>, <netdb.h>,
 <netinet/in.h>, and <sys/socket.h>. An implementation may search this library in
 the absence of this operand.
- 8350 **-l y** This operand shall make visible all functions required by the C-language output of
 yacc that are not made available through the -l c operand.

8352 In the absence of options that inhibit invocation of the link editor, such as `-c` or `-E`, the `c99` utility
 8353 shall cause the equivalent of a `-l c` operand to be passed to the link editor as the last `-l` operand,
 8354 causing it to be searched after all other object files and libraries are loaded.

8355 It is unspecified whether the libraries `libc.a`, `libm.a`, `librt.a`, `libpthread.a`, `libl.a`, `liby.a`, or `libxnet`
 8356 exist as regular files. The implementation may accept as `-l` operands names of objects that do
 8357 not exist as regular files.

8358 External Symbols

8359 The C compiler and link editor shall support the significance of external symbols up to a length
 8360 of at least 31 bytes; the action taken upon encountering symbols exceeding the implementation-
 8361 defined maximum symbol length is unspecified.

8362 The compiler and link editor shall support a minimum of 511 external symbols per source or
 8363 object file, and a minimum of 4 095 external symbols in total. A diagnostic message shall be
 8364 written to the standard output if the implementation-defined limit is exceeded; other actions are
 8365 unspecified.

8366 Programming Environments

8367 All implementations shall support one of the following programming environments as a default.
 8368 Implementations may support more than one of the following programming environments.
 8369 Applications can use `sysconf()` or `getconf` to determine which programming environments are
 8370 supported.

8371 **Table 4-4** Programming Environments: Type Sizes

8372 Programming Environment 8373 <i>getconf Name</i>	8372 Bits in 8373 int	8372 Bits in 8373 long	8372 Bits in 8373 pointer	8372 Bits in 8373 off_t
8374 <code>_POSIX_V6_ILP32_OFF32</code>	32	32	32	32
8375 <code>_POSIX_V6_ILP32_OFFBIG</code>	32	32	32	≥ 64
8376 <code>_POSIX_V6_LP64_OFF64</code>	32	64	64	64
8377 <code>_POSIX_V6_LPBIG_OFFBIG</code>	≥ 32	≥ 64	≥ 64	≥ 64

8378 All implementations shall support one or more environments where the widths of the following
 8379 types are no greater than the width of type `long`:

8380 `blksize_t`, `cc_t`, `mode_t`, `nfds_t`, `pid_t`, `ptrdiff_t`, `size_t`, `speed_t`, `ssize_t`, `suseconds_t`,
 8381 `tcflag_t`, `useconds_t`, `wchar_t`, `wint_t`

8382 The executable files created when these environments are selected shall be in a proper format for
 8383 execution by the `exec` family of functions. Each environment may be one of the ones in Table 4-4,
 8384 or it may be another environment. The names for the environments that meet this requirement
 8385 shall be output by a `getconf` command using the `_POSIX_V6_WIDTH_RESTRICTED_ENVS`
 8386 argument. If more than one environment meets the requirement, the names of all such
 8387 environments shall be output on separate lines. Any of these names can then be used in a
 8388 subsequent `getconf` command to obtain the flags specific to that environment with the following
 8389 suffixes added as appropriate:

8390 `_CFLAGS` To get the C compiler flags.

8391 `_LDFLAGS` To get the linker/loader flags.

8392 `_LIBS` To get the libraries.

8393 This requirement may be removed in a future version of IEEE Std 1003.1.

When this utility processes a file containing a function called *main()*, it shall be defined with a return type equivalent to **int**. Using *return* from the initial call to *main()* shall be equivalent (other than with respect to language scope issues) to calling *exit()* with the returned value. Reaching the end of the initial call to *main()* shall be equivalent to calling *exit(0)*. The implementation shall not declare a prototype for this function.

Implementations provide configuration strings for C compiler flags, linker/loader flags, and libraries for each supported environment. When an application needs to use a specific programming environment rather than the implementation default programming environment while compiling, the application shall first verify that the implementation supports the desired environment. If the desired programming environment is supported, the application shall then invoke *c99* with the appropriate C compiler flags as the first options for the compile, the appropriate linker/loader flags after any other options but before any operands, and the appropriate libraries at the end of the operands.

Conforming applications shall not attempt to link together object files compiled for different programming models. Applications shall also be aware that binary data placed in shared memory or in files might not be recognized by applications built for other programming models.

Table 4-5 Programming Environments: *c99* and *cc* Arguments

Programming Environment <i>getconf Name</i>	Use	<i>c99</i> and <i>cc</i> Arguments <i>getconf Name</i>
_POSIX_V6_ILP32_OFF32	C Compiler Flags Linker/Loader Flags Libraries	POSIX_V6_ILP32_OFF32_CFLAGS POSIX_V6_ILP32_OFF32_LDFLAGS POSIX_V6_ILP32_OFF32_LIBS
_POSIX_V6_ILP32_OFFBIG	C Compiler Flags Linker/Loader Flags Libraries	POSIX_V6_ILP32_OFFBIG_CFLAGS POSIX_V6_ILP32_OFFBIG_LDFLAGS POSIX_V6_ILP32_OFFBIG_LIBS
_POSIX_V6_LP64_OFF64	C Compiler Flags Linker/Loader Flags Libraries	POSIX_V6_LP64_OFF64_CFLAGS POSIX_V6_LP64_OFF64_LDFLAGS POSIX_V6_LP64_OFF64_LIBS
_POSIX_V6_LPBIG_OFFBIG	C Compiler Flags Linker/Loader Flags Libraries	POSIX_V6_LPBIG_OFFBIG_CFLAGS POSIX_V6_LPBIG_OFFBIG_LDFLAGS POSIX_V6_LPBIG_OFFBIG_LIBS

EXIT STATUS

The following exit values shall be returned:

0 Successful compilation or link edit.

>0 An error occurred.

CONSEQUENCES OF ERRORS

When *c99* encounters a compilation error that causes an object file not to be created, it shall write a diagnostic to standard error and continue to compile other source code operands, but it shall not perform the link phase and return a non-zero exit status. If the link edit is unsuccessful, a diagnostic message shall be written to standard error and *c99* exits with a non-zero status. A conforming application shall rely on the exit status of *c99*, rather than on the existence or mode of the executable file.

8436 **APPLICATION USAGE**

8437 Since the *c99* utility usually creates files in the current directory during the compilation process,
8438 it is typically necessary to run the *c99* utility in a directory in which a file can be created.

8439 On systems providing POSIX Conformance (see the Base Definitions volume of
8440 IEEE Std 1003.1-2001, Chapter 2, Conformance), *c99* is required only with the C-Language
8441 Development option; XSI-conformant systems always provide *c99*.

8442 Some historical implementations have created .o files when –c is not specified and more than
8443 one source file is given. Since this area is left unspecified, the application cannot rely on .o files
8444 being created, but it also must be prepared for any related .o files that already exist being deleted
8445 at the completion of the link edit.

8446 Some historical implementations have permitted –L options to be interspersed with –I operands
8447 on the command line. For an application to compile consistently on systems that do not behave
8448 like this, it is necessary for a conforming application to supply all –L options before any of the –I
8449 options.

8450 There is the possible implication that if a user supplies versions of the standard functions (before
8451 they would be encountered by an implicit –I c or explicit –I m), that those versions would be
8452 used in place of the standard versions. There are various reasons this might not be true
8453 (functions defined as macros, manipulations for clean name space, and so on), so the existence of
8454 files named in the same manner as the standard libraries within the –L directories is explicitly
8455 stated to produce unspecified behavior.

8456 All of the functions specified in the System Interfaces volume of IEEE Std 1003.1-2001 may be
8457 made visible by implementations when the Standard C Library is searched. Conforming
8458 applications must explicitly request searching the other standard libraries when functions made
8459 visible by those libraries are used.

8460 **EXAMPLES**

- 8461 1. The following usage example compiles **foo.c** and creates the executable file **foo**:

8462 *c99 -o foo foo.c*

8463 The following usage example compiles **foo.c** and creates the object file **foo.o**:

8464 *c99 -c foo.c*

8465 The following usage example compiles **foo.c** and creates the executable file **a.out**:

8466 *c99 foo.c*

8467 The following usage example compiles **foo.c**, links it with **bar.o**, and creates the executable
8468 file **a.out**. It may also create and leave **foo.o**:

8469 *c99 foo.c bar.o*

- 8470 2. The following example shows how an application using threads interfaces can test for
8471 support of and use a programming environment supporting 32-bit **int**, **long**, and **pointer**
8472 types and an **off_t** type using at least 64 bits:

```
8473 if [ $(getconf _POSIX_V6_ILP32_OFFBIG) != "-1" ]  
8474 then  
8475     c99 $(getconf POSIX_V6_ILP32_OFFBIG_CFLAGS) -D_XOPEN_SOURCE=600 \  
8476         $(getconf POSIX_V6_ILP32_OFFBIG_LDFLAGS) foo.c -o foo \  
8477             $(getconf POSIX_V6_ILP32_OFFBIG_LIBS) -l pthread  
8478 else  
8479     echo ILP32_OFFBIG programming environment not supported
```

```
8480         exit 1
8481     fi
```

3. The following examples clarify the use and interactions of **-L** options and **-l** operands.

8483 Consider the case in which module **a.c** calls function *f()* in library **libQ.a**, and module **b.c**
 8484 calls function *g()* in library **libp.a**. Assume that both libraries reside in **/a/b/c**. The
 8485 command line to compile and link in the desired way is:

```
8486 c99 -L /a/b/c main.o a.c -l Q b.c -l p
```

8487 In this case the **-l Q** operand need only precede the first **-l p** operand, since both **libQ.a**
 8488 and **libp.a** reside in the same directory.

8489 Multiple **-L** operands can be used when library name collisions occur. Building on the
 8490 previous example, suppose that the user wants to use a new **libp.a**, in **/a/a/a**, but still wants
 8491 *f()* from **/a/b/c/libQ.a**:

```
8492 c99 -L /a/a/a -L /a/b/c main.o a.c -l Q b.c -l p
```

8493 In this example, the linker searches the **-L** options in the order specified, and finds
 8494 **/a/a/a/libp.a** before **/a/b/c/libp.a** when resolving references for **b.c**. The order of the **-l**
 8495 operands is still important, however.

4. The following example shows how an application can use a programming environment where the widths of the following types:

```
8498     blksize_t, cc_t, mode_t, nfds_t, pid_t, ptrdiff_t, size_t, speed_t, ssize_t, suseconds_t,
8499     tcflag_t, useconds_t, wchar_t, wint_t
```

8500 are no greater than the width of type **long**:

```
8501 # First choose one of the listed environments ...
8502 # ... if there are no additional constraints, the first one will do:
8503 CENV=$(getconf _POSIX_V6_WIDTH_RESTRICTED_ENVS | head -n 1)
8504 # ... or, if an environment that supports large files is preferred,
8505 # look for names that contain "OFF64" or "OFFBIG". (This chooses
8506 # the last one in the list if none match.)
8507 for CENV in $(getconf _POSIX_V6_WIDTH_RESTRICTED_ENVS)
8508 do
8509     case $CENV in
8510         *OFF64* | *OFFBIG*) break ;;
8511     esac
8512 done
8513 # The chosen environment name can now be used like this:
8514 c99 ${CENV}_CFLAGS -D _POSIX_C_SOURCE=200112L \
8515 ${CENV}_LDFLAGS foo.c -o foo \
8516 ${CENV}_LIBS
```

8517 RATIONALE

8518 The *c99* utility is based on the *c89* utility originally introduced in the ISO POSIX-2:1993 standard.

8519 Some of the changes from *c89* include the modification to the contents of the Standard Libraries
 8520 section to account for new headers and options; for example, **<spawn.h>** added to the **-l rt**
 8521 operand, and the **-l trace** operand added for the Tracing functions.

8522 FUTURE DIRECTIONS

8523 None.

8524 SEE ALSO

8525 Section 1.7.1.4 (on page 4), *ar*, *getconf*, *make*, *nm*, *strip*, *umask*, the System Interfaces volume of
8526 IEEE Std 1003.1-2001, *exec*, *sysconf()*, the Base Definitions volume of IEEE Std 1003.1-2001,
8527 Chapter 13, Headers

8528 CHANGE HISTORY

8529 First released in Issue 6. Included for alignment with the ISO/IEC 9899:1999 standard.

8530 NAME

8531 cal — print a calendar

8532 SYNOPSIS

8533 XSI `cal [[month] year]`

8534

8535 DESCRIPTION

8536 The *cal* utility shall write a calendar to standard output using the Julian calendar for dates from
8537 January 1, 1 through September 2, 1752 and the Gregorian calendar for dates from September 14,
8538 1752 through December 31, 9999 as though the Gregorian calendar had been adopted on
8539 September 14, 1752.

8540 OPTIONS

8541 None.

8542 OPERANDS

8543 The following operands shall be supported:

8544 *month* Specify the month to be displayed, represented as a decimal integer from 1
8545 (January) to 12 (December). The default shall be the current month.

8546 *year* Specify the year for which the calendar is displayed, represented as a decimal
8547 integer from 1 to 9999. The default shall be the current year.

8548 STDIN

8549 Not used.

8550 INPUT FILES

8551 None.

8552 ENVIRONMENT VARIABLES

8553 The following environment variables shall affect the execution of *cal*:

8554 *LANG* Provide a default value for the internationalization variables that are unset or null.
8555 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
8556 Internationalization Variables for the precedence of internationalization variables
8557 used to determine the values of locale categories.)

8558 *LC_ALL* If set to a non-empty string value, override the values of all the other
8559 internationalization variables.

8560 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
8561 characters (for example, single-byte as opposed to multi-byte characters in
8562 arguments).

8563 *LC_MESSAGES*

8564 Determine the locale that should be used to affect the format and contents of
8565 diagnostic messages written to standard error, and informative messages written
8566 to standard output.

8567 *LC_TIME* Determine the format and contents of the calendar.

8568 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

8569 *TZ* Determine the timezone used to calculate the value of the current month.

8570 ASYNCHRONOUS EVENTS

8571 Default.

8572 STDOUT

8573 The standard output shall be used to display the calendar, in an unspecified format.

8574 STDERR

8575 The standard error shall be used only for diagnostic messages.

8576 OUTPUT FILES

8577 None.

8578 EXTENDED DESCRIPTION

8579 None.

8580 EXIT STATUS

8581 The following exit values shall be returned:

8582 0 Successful completion.

8583 >0 An error occurred.

8584 CONSEQUENCES OF ERRORS

8585 Default.

8586 APPLICATION USAGE

8587 Note that:

8588 cal 83

8589 refers to A.D. 83, not 1983.

8590 EXAMPLES

8591 None.

8592 RATIONALE

8593 None.

8594 FUTURE DIRECTIONS

8595 A future version of IEEE Std 1003.1-2001 may support locale-specific recognition of the date of
8596 adoption of the Gregorian calendar.

8597 SEE ALSO

8598 None.

8599 CHANGE HISTORY

8600 First released in Issue 2.

8601 Issue 6

8602 The DESCRIPTION is updated to allow for traditional behavior for years before the adoption of
8603 the Gregorian calendar.

8604 NAME

8605 cat — concatenate and print files

8606 SYNOPSIS

8607 cat [-u][*file* ...]

8608 DESCRIPTION

8609 The *cat* utility shall read files in sequence and shall write their contents to the standard output in
8610 the same sequence.

8611 OPTIONS

8612 The *cat* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
8613 Utility Syntax Guidelines.

8614 The following option shall be supported:

8615 **-u** Write bytes from the input file to the standard output without delay as each is
8616 read.

8617 OPERANDS

8618 The following operand shall be supported:

8619 *file* A pathname of an input file. If no *file* operands are specified, the standard input
8620 shall be used. If a *file* is ‘-’, the *cat* utility shall read from the standard input at
8621 that point in the sequence. The *cat* utility shall not close and reopen standard input
8622 when it is referenced in this way, but shall accept multiple occurrences of ‘-’ as a
8623 *file* operand.

8624 STDIN

8625 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is ‘-’.
8626 See the INPUT FILES section.

8627 INPUT FILES

8628 The input files can be any file type.

8629 ENVIRONMENT VARIABLES

8630 The following environment variables shall affect the execution of *cat*:

8631 **LANG** Provide a default value for the internationalization variables that are unset or null.
8632 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
8633 Internationalization Variables for the precedence of internationalization variables
8634 used to determine the values of locale categories.)

8635 **LC_ALL** If set to a non-empty string value, override the values of all the other
8636 internationalization variables.

8637 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
8638 characters (for example, single-byte as opposed to multi-byte characters in
8639 arguments).

8640 LC_MESSAGES

8641 Determine the locale that should be used to affect the format and contents of
8642 diagnostic messages written to standard error.

8643 **xsi NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

8644 ASYNCHRONOUS EVENTS

8645 Default.

8646 STDOUT

8647 The standard output shall contain the sequence of bytes read from the input files. Nothing else
8648 shall be written to the standard output.

8649 STDERR

8650 The standard error shall be used only for diagnostic messages.

8651 OUTPUT FILES

8652 None.

8653 EXTENDED DESCRIPTION

8654 None.

8655 EXIT STATUS

8656 The following exit values shall be returned:

8657 0 All input files were output successfully.

8658 >0 An error occurred.

8659 CONSEQUENCES OF ERRORS

8660 Default.

8661 APPLICATION USAGE

8662 The **-u** option has value in prototyping non-blocking reads from FIFOs. The intent is to support
8663 the following sequence:

```
8664       mkfifo foo  
8665       cat -u foo > /dev/tty13 &  
8666       cat -u > foo
```

8667 It is unspecified whether standard output is or is not buffered in the default case. This is
8668 sometimes of interest when standard output is associated with a terminal, since buffering may
8669 delay the output. The presence of the **-u** option guarantees that unbuffered I/O is available. It is
8670 implementation-defined whether the **cat** utility buffers output if the **-u** option is not specified.
8671 Traditionally, the **-u** option is implemented using the equivalent of the **setvbuf()** function
8672 defined in the System Interfaces volume of IEEE Std 1003.1-2001.

8673 EXAMPLES

8674 The following command:

```
8675       cat myfile
```

8676 writes the contents of the file **myfile** to standard output.

8677 The following command:

```
8678       cat doc1 doc2 > doc.all
```

8679 concatenates the files **doc1** and **doc2** and writes the result to **doc.all**.

8680 Because of the shell language mechanism used to perform output redirection, a command such
8681 as this:

```
8682       cat doc doc.end > doc
```

8683 causes the original data in **doc** to be lost.

8684 The command:

```
8685       cat start - middle - end > file
```

8686 when standard input is a terminal, gets two arbitrary pieces of input from the terminal with a
8687 single invocation of *cat*. Note, however, that if standard input is a regular file, this would be
8688 equivalent to the command:

8689 *cat start - middle /dev/null end > file*

8690 because the entire contents of the file would be consumed by *cat* the first time '*-*' was used as a
8691 *file* operand and an end-of-file condition would be detected immediately when '*-*' was
8692 referenced the second time.

8693 **RATIONALE**

8694 Historical versions of the *cat* utility include the options **-e**, **-t**, and **-v**, which permit the ends of
8695 lines, *<tab>*s, and invisible characters, respectively, to be rendered visible in the output. The
8696 standard developers omitted these options because they provide too fine a degree of control
8697 over what is made visible, and similar output can be obtained using a command such as:

8698 *sed -n -e 's/\$/\$/' -e l pathname*

8699 The **-s** option was omitted because it corresponds to different functions in BSD and System V-
8700 based systems. The BSD **-s** option to squeeze blank lines can be accomplished by the shell script
8701 shown in the following example:

```
8702 sed -n '
8703 # Write non-empty lines.
8704 ./ {
8705   p
8706   d
8707 }
8708 # Write a single empty line, then look for more empty lines.
8709 /^$/ p
8710 # Get next line, discard the held <newline> (empty line),
8711 # and look for more empty lines.
8712 :Empty
8713 /^$/ {
8714   N
8715   s/.*/
8716   b Empty
8717 }
8718 # Write the non-empty line before going back to search
8719 # for the first in a set of empty lines.
8720   p
8721 '
```

8722 The System V **-s** option to silence error messages can be accomplished by redirecting the
8723 standard error. Note that the BSD documentation for *cat* uses the term "blank line" to mean the
8724 same as the POSIX "empty line": a line consisting only of a *<newline>*.

8725 The BSD **-n** option was omitted because similar functionality can be obtained from the **-n**
8726 option of the *pr* utility.

8727 **FUTURE DIRECTIONS**

8728 None.

8729 **SEE ALSO**

8730 *more*, the System Interfaces volume of IEEE Std 1003.1-2001, *setvbuf()*

8731 **CHANGE HISTORY**

8732 First released in Issue 2.

8733 NAME

8734 cd — change the working directory

8735 SYNOPSIS

8736 cd [-L] [-P] [directory]

8737 cd -

8738 DESCRIPTION

8739 The *cd* utility shall change the working directory of the current shell execution environment (see
8740 Section 2.12 (on page 61)) by executing the following steps in sequence. (In the following steps,
8741 the symbol **curpath** represents an intermediate value used to simplify the description of the
8742 algorithm used by *cd*. There is no requirement that **curpath** be made visible to the application.)

- 8743 1. If no *directory* operand is given and the *HOME* environment variable is empty or
8744 undefined, the default behavior is implementation-defined and no further steps shall be
8745 taken.
- 8746 2. If no *directory* operand is given and the *HOME* environment variable is set to a non-empty
8747 value, the *cd* utility shall behave as if the directory named in the *HOME* environment
8748 variable was specified as the *directory* operand.
- 8749 3. If the *directory* operand begins with a slash character, set **curpath** to the operand and
8750 proceed to step 7.
- 8751 4. If the first component of the *directory* operand is dot or dot-dot, proceed to step 6.
- 8752 5. Starting with the first pathname in the colon-separated pathnames of *CDPATH* (see the
8753 ENVIRONMENT VARIABLES section) if the pathname is non-null, test if the
8754 concatenation of that pathname, a slash character, and the *directory* operand names a
8755 directory. If the pathname is null, test if the concatenation of dot, a slash character, and the
8756 operand names a directory. In either case, if the resulting string names an existing
8757 directory, set **curpath** to that string and proceed to step 7. Otherwise, repeat this step with
8758 the next pathname in *CDPATH* until all pathnames have been tested.
- 8759 6. Set **curpath** to the string formed by the concatenation of the value of *PWD*, a slash
8760 character, and the operand.
- 8761 7. If the **-P** option is in effect, the *cd* utility shall perform actions equivalent to the *chdir()*
8762 function, called with **curpath** as the *path* argument. If these actions succeed, the *PWD*
8763 environment variable shall be set to an absolute pathname for the current working
8764 directory and shall not contain filename components that, in the context of pathname
8765 resolution, refer to a file of type symbolic link. If there is insufficient permission on the new
8766 directory, or on any parent of that directory, to determine the current working directory,
8767 the value of the *PWD* environment variable is unspecified. If the actions equivalent to
8768 *chdir()* fail for any reason, the *cd* utility shall display an appropriate error message and not
8769 alter the *PWD* environment variable. Whether the actions equivalent to *chdir()* succeed or
8770 fail, no further steps shall be taken.
- 8771 8. The **curpath** value shall then be converted to canonical form as follows, considering each
8772 component from beginning to end, in sequence:
 - 8773 a. Dot components and any slashes that separate them from the next component shall
8774 be deleted.
 - 8775 b. For each dot-dot component, if there is a preceding component and it is neither root
8776 nor dot-dot, the preceding component, all slashes separating the preceding
8777 component from dot-dot, dot-dot and all slashes separating dot-dot from the
8778 following component shall be deleted.

- 8779 c. An implementation may further simplify **curpath** by removing any trailing slash
8780 characters that are not also leading slashes, replacing multiple non-leading
8781 consecutive slashes with a single slash, and replacing three or more leading slashes
8782 with a single slash. If, as a result of this canonicalization, the **curpath** variable is null,
8783 no further steps shall be taken.
- 8784 9. The *cd* utility shall then perform actions equivalent to the *chdir()* function called with
8785 **curpath** as the *path* argument. If these actions failed for any reason, the *cd* utility shall
8786 display an appropriate error message and no further steps shall be taken. The *PWD*
8787 environment variable shall be set to **curpath**.

8788 If, during the execution of the above steps, the *PWD* environment variable is changed, the
8789 *OLDPWD* environment variable shall also be changed to the value of the old working directory
8790 (that is the current working directory immediately prior to the call to *cd*).

8791 OPTIONS

8792 The *cd* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
8793 Utility Syntax Guidelines.

8794 The following options shall be supported by the implementation:

- 8795 **-L** Handle the operand dot-dot logically; symbolic link components shall not be
8796 resolved before dot-dot components are processed (see steps 8. and 9. in the
8797 DESCRIPTION).
- 8798 **-P** Handle the operand dot-dot physically; symbolic link components shall be
8799 resolved before dot-dot components are processed (see step 7. in the
8800 DESCRIPTION).

8801 If both **-L** and **-P** options are specified, the last of these options shall be used and all others
8802 ignored. If neither **-L** nor **-P** is specified, the operand shall be handled dot-dot logically; see the
8803 DESCRIPTION.

8804 OPERANDS

8805 The following operands shall be supported:

8806 **directory** An absolute or relative pathname of the directory that shall become the new
8807 working directory. The interpretation of a relative pathname by *cd* depends on the
8808 **-L** option and the *CDPATH* and *PWD* environment variables. If *directory* is an
8809 empty string, the results are unspecified.

8810 **-** When a hyphen is used as the operand, this shall be equivalent to the command:

8811 `cd "$OLDPWD" && pwd`

8812 which changes to the previous working directory and then writes its name.

8813 STDIN

8814 Not used.

8815 INPUT FILES

8816 None.

8817 ENVIRONMENT VARIABLES

8818 The following environment variables shall affect the execution of *cd*:

8819 **CDPATH** A colon-separated list of pathnames that refer to directories. The *cd* utility shall use
8820 this list in its attempt to change the directory, as described in the DESCRIPTION.
8821 An empty string in place of a directory pathname represents the current directory.
8822 If *CDPATH* is not set, it shall be treated as if it were an empty string.

8823	<i>HOME</i>	The name of the directory, used when no <i>directory</i> operand is specified.
8824	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
8828	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
8830	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
8833	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
8836 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
8837	<i>OLDPWD</i>	A pathname of the previous working directory, used by <i>cd -</i> .
8838	<i>PWD</i>	This variable shall be set as specified in the DESCRIPTION. If an application sets or unsets the value of <i>PWD</i> , the behavior of <i>cd</i> is unspecified.
8840	ASYNCHRONOUS EVENTS	
8841		Default.
8842	STDOUT	
8843		If a non-empty directory name from <i>CDPATH</i> is used, or if <i>cd -</i> is used, an absolute pathname of the new working directory shall be written to the standard output as follows:
8845		"%s\n", <new directory>
8846		Otherwise, there shall be no output.
8847	STDERR	
8848		The standard error shall be used only for diagnostic messages.
8849	OUTPUT FILES	
8850		None.
8851	EXTENDED DESCRIPTION	
8852		None.
8853	EXIT STATUS	
8854		The following exit values shall be returned:
8855		0 The directory was successfully changed.
8856		>0 An error occurred.
8857	CONSEQUENCES OF ERRORS	
8858		The working directory shall remain unchanged.

8859 APPLICATION USAGE

8860 Since *cd* affects the current shell execution environment, it is always provided as a shell regular
8861 built-in. If it is called in a subshell or separate utility execution environment, such as one of the
8862 following:

```
8863 ( cd /tmp )
8864 nohup cd
8865 find . -exec cd {} \;
```

8866 it does not affect the working directory of the caller's environment.

8867 The user must have execute (search) permission in *directory* in order to change to it.

8868 EXAMPLES

8869 None.

8870 RATIONALE

8871 The use of the *CDPATH* was introduced in the System V shell. Its use is analogous to the use of the
8872 *PATH* variable in the shell. The BSD C shell used a shell parameter *cpath* for this purpose.

8873 A common extension when *HOME* is undefined is to get the login directory from the user
8874 database for the invoking user. This does not occur on System V implementations.

8875 Some historical shells, such as the KornShell, took special actions when the directory name
8876 contained a dot-dot component, selecting the logical parent of the directory, rather than the
8877 actual parent directory; that is, it moved up one level toward the ‘‘/’’ in the pathname,
8878 remembering what the user typed, rather than performing the equivalent of:

```
8879 chdir("../");
```

8880 In such a shell, the following commands would not necessarily produce equivalent output for all
8881 directories:

```
8882 cd .. && ls      ls ..
```

8883 This behavior is now the default. It is not consistent with the definition of dot-dot in most
8884 historical practice; that is, while this behavior has been optionally available in the KornShell,
8885 other shells have historically not supported this functionality. The logical pathname is stored in
8886 the *PWD* environment variable when the *cd* utility completes and this value is used to construct
8887 the next directory name if *cd* is invoked with the *-L* option.

8888 FUTURE DIRECTIONS

8889 None.

8890 SEE ALSO

8891 Section 2.12 (on page 61), *pwd*, the System Interfaces volume of IEEE Std 1003.1-2001, *chdir()*

8892 CHANGE HISTORY

8893 First released in Issue 2.

8894 Issue 6

8895 The following new requirements on POSIX implementations derive from alignment with the
8896 Single UNIX Specification:

- 8897 • The *cd* – operand, *PWD*, and *OLDPWD* are added.

8898 The *-L* and *-P* options are added to align with the IEEE P1003.2b draft standard. This also
8899 includes the introduction of a new description to include the effect of these options.

8900 NAME

8901 cflow — generate a C-language flowgraph (**DEVELOPMENT**)

8902 SYNOPSIS

8903 XSI cflow [-r][-d num][-D name[=def]] ... [-i incl][-I dir] ...
8904 [-U dir] ... file ...

8905

8906 DESCRIPTION

8907 The *cflow* utility shall analyze a collection of object files or assembler, C-language, *lex*, or *yacc*
8908 source files, and attempt to build a graph, written to standard output, charting the external
8909 references.

8910 OPTIONS

8911 The *cflow* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
8912 12.2, Utility Syntax Guidelines, except that the order of the **-D**, **-I**, and **-U** options (which are
8913 identical to their interpretation by *c99*) is significant.

8914 The following options shall be supported:

8915 **-d num** Indicate the depth at which the flowgraph is cut off. The application shall ensure
8916 that the argument *num* is a decimal integer. By default this is a very large number
8917 (typically greater than 32 000). Attempts to set the cut-off depth to a non-positive
8918 integer shall be ignored.

8919 **-i incl** Increase the number of included symbols. The *incl* option-argument is one of the
8920 following characters:

8921 **x** Include external and static data symbols. The default shall be to include only
8922 functions in the flowgraph.
8923 **_** (Underscore) Include names that begin with an underscore. The default shall
8924 be to exclude these functions (and data if **-i x** is used).

8925 **-r** Reverse the caller:callee relationship, producing an inverted listing showing the
8926 callers of each function. The listing shall also be sorted in lexicographical order by
8927 callee.

8928 OPERANDS

8929 The following operand is supported:

8930 **file** The pathname of a file for which a graph is to be generated. Filenames suffixed by
8931 **.l** shall shall be taken to be *lex* input, **.y** as *yacc* input, **.c** as *c99* input, and **.i** as the
8932 output of *c99 -E*. Such files shall be processed as appropriate, determined by their
8933 suffix.

8934 Files suffixed by **.s** (conventionally assembler source) may have more limited
8935 information extracted from them.

8936 STDIN

8937 Not used.

8938 INPUT FILES

8939 The input files shall be object files or assembler, C-language, *lex*, or *yacc* source files.

8940 ENVIRONMENT VARIABLES

8941 The following environment variables shall affect the execution of *cflow*:

8942 **LANG** Provide a default value for the internationalization variables that are unset or null.
8943 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,

8944 Internationalization Variables for the precedence of internationalization variables
8945 used to determine the values of locale categories.)

8946 *LC_ALL* If set to a non-empty string value, override the values of all the other
8947 internationalization variables.

8948 *LC_COLLATE*
8949 Determine the locale for the ordering of the output when the **-r** option is used.

8950 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
8951 characters (for example, single-byte as opposed to multi-byte characters in
8952 arguments and input files).

8953 *LC_MESSAGES*
8954 Determine the locale that should be used to affect the format and contents of
8955 diagnostic messages written to standard error.

8956 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

8957 **ASYNCHRONOUS EVENTS**

8958 Default.

8959 **STDOUT**

8960 The flowgraph written to standard output shall be formatted as follows:

8961 "*%d %s:%s\n*", *<reference number>*, *<global>*, *<definition>*

8962 Each line of output begins with a reference (that is, line) number, followed by indentation of at
8963 least one column position per level. This is followed by the name of the global, a colon, and its
8964 definition. Normally globals are only functions not defined as an external or beginning with an
8965 underscore; see the OPTIONS section for the **-i** inclusion option. For information extracted from
8966 C-language source, the definition consists of an abstract type declaration (for example, **char ***)
8967 and, delimited by angle brackets, the name of the source file and the line number where the
8968 definition was found. Definitions extracted from object files indicate the filename and location
8969 counter under which the symbol appeared (for example, *text*).

8970 Once a definition of a name has been written, subsequent references to that name contain only
8971 the reference number of the line where the definition can be found. For undefined references,
8972 only "*< >*" shall be written.

8973 **STDERR**

8974 The standard error shall be used only for diagnostic messages.

8975 **OUTPUT FILES**

8976 None.

8977 **EXTENDED DESCRIPTION**

8978 None.

8979 **EXIT STATUS**

8980 The following exit values shall be returned:

8981 0 Successful completion.

8982 >0 An error occurred.

8983 **CONSEQUENCES OF ERRORS**

8984 Default.

8985 **APPLICATION USAGE**

8986 Files produced by *lex* and *yacc* cause the reordering of line number declarations, and this can
8987 confuse *cflow*. To obtain proper results, the input of *yacc* or *lex* must be directed to *cflow*.

8988 **EXAMPLES**

8989 Given the following in **file.c**:

```
8990       int i;
8991       int f();
8992       int g();
8993       int h();
8994       int
8995       main()
8996       {
8997           f();
8998           g();
8999           f();
9000       }
9001       int
9002       f()
9003       {
9004           i = h();
9005       }
```

9006 The command:

```
9007       cflow -i x file.c
```

9008 produces the output:

```
9009       1 main: int(), <file.c 6>
9010       2     f: int(), <file.c 13>
9011       3       h: <>
9012       4        i: int, <file.c 1>
9013       5     g: <>
```

9014 **RATIONALE**

9015 None.

9016 **FUTURE DIRECTIONS**

9017 None.

9018 **SEE ALSO**

9019 *c99*, *lex*, *yacc*

9020 **CHANGE HISTORY**

9021 First released in Issue 2.

9022 **Issue 6**

9023 The normative text is reworded to avoid use of the term “must” for application requirements.

9024 **NAME**

9025 chgrp — change the file group ownership

9026 **SYNOPSIS**9027 chgrp -hR *group file* ...9028 chgrp -R [-H | -L | -P] *group file* ...9029 **DESCRIPTION**9030 The *chgrp* utility shall set the group ID of the file named by each *file* operand to the group ID specified by the *group* operand.9032 For each *file* operand, or, if the **-R** option is used, each file encountered while walking the
9033 directory trees specified by the *file* operands, the *chgrp* utility shall perform actions equivalent to
9034 the *chown()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001, called
9035 with the following arguments:

- The *file* operand shall be used as the *path* argument.
- The user ID of the file shall be used as the *owner* argument.
- The specified group ID shall be used as the *group* argument.

9039 Unless *chgrp* is invoked by a process with appropriate privileges, the set-user-ID and set-group-ID bits of a regular file shall be cleared upon successful completion; the set-user-ID and set-group-ID bits of other file types may be cleared.9042 **OPTIONS**9043 The *chgrp* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
9044 12.2, Utility Syntax Guidelines.

9045 The following options shall be supported by the implementation:

- h If the system supports group IDs for symbolic links, for each *file* operand that names a file of type symbolic link, *chgrp* shall attempt to set the group ID of the symbolic link instead of the file referenced by the symbolic link. If the system does not support group IDs for symbolic links, for each *file* operand that names a file of type symbolic link, *chgrp* shall do nothing more with the current file and shall go on to any remaining files.
- H If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line, *chgrp* shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.
- L If the **-R** option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, *chgrp* shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.
- P If the **-R** option is specified and a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, *chgrp* shall change the group ID of the symbolic link if the system supports this operation. The *chgrp* utility shall not follow the symbolic link to any other part of the file hierarchy.
- R Recursively change file group IDs. For each *file* operand that names a directory, *chgrp* shall change the group of the directory and all files in the file hierarchy below it. Unless a **-H**, **-L**, or **-P** option is specified, it is unspecified which of these options will be used as the default.

9067 Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be
9068 considered an error. The last option specified shall determine the behavior of the utility.

9069 OPERANDS

9070 The following operands shall be supported:

9071 *group* A group name from the group database or a numeric group ID. Either specifies a
9072 group ID to be given to each file named by one of the *file* operands. If a numeric
9073 *group* operand exists in the group database as a group name, the group ID number
9074 associated with that group name is used as the group ID.

9075 *file* A pathname of a file whose group ID is to be modified.

9076 STDIN

9077 Not used.

9078 INPUT FILES

9079 None.

9080 ENVIRONMENT VARIABLES

9081 The following environment variables shall affect the execution of *chgrp*:

9082 *LANG* Provide a default value for the internationalization variables that are unset or null.
9083 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
9084 Internationalization Variables for the precedence of internationalization variables
9085 used to determine the values of locale categories.)

9086 *LC_ALL* If set to a non-empty string value, override the values of all the other
9087 internationalization variables.

9088 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
9089 characters (for example, single-byte as opposed to multi-byte characters in
9090 arguments).

9091 *LC_MESSAGES*

9092 Determine the locale that should be used to affect the format and contents of
9093 diagnostic messages written to standard error.

9094 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

9095 ASYNCHRONOUS EVENTS

9096 Default.

9097 STDOUT

9098 Not used.

9099 STDERR

9100 The standard error shall be used only for diagnostic messages.

9101 OUTPUT FILES

9102 None.

9103 EXTENDED DESCRIPTION

9104 None.

9105 EXIT STATUS

9106 The following exit values shall be returned:

9107 0 The utility executed successfully and all requested changes were made.

9108 >0 An error occurred.

9109 CONSEQUENCES OF ERRORS

9110 Default.

9111 APPLICATION USAGE

9112 Only the owner of a file or the user with appropriate privileges may change the owner or group
9113 of a file.

9114 Some implementations restrict the use of *chgrp* to a user with appropriate privileges when the
9115 *group* specified is not the effective group ID or one of the supplementary group IDs of the calling
9116 process.

9117 EXAMPLES

9118 None.

9119 RATIONALE

9120 The System V and BSD versions use different exit status codes. Some implementations used the
9121 exit status as a count of the number of errors that occurred; this practice is unworkable since it
9122 can overflow the range of valid exit status values. The standard developers chose to mask these
9123 by specifying only 0 and >0 as exit values.

9124 The functionality of *chgrp* is described substantially through references to *chown()*. In this way,
9125 there is no duplication of effort required for describing the interactions of permissions, multiple
9126 groups, and so on.

9127 FUTURE DIRECTIONS

9128 None.

9129 SEE ALSO

9130 *chmod*, *chown*, the System Interfaces volume of IEEE Std 1003.1-2001, *chown()*

9131 CHANGE HISTORY

9132 First released in Issue 2.

9133 Issue 6

9134 New options **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These
9135 options affect the processing of symbolic links.

9136 IEEE PASC Interpretation 1003.2 #172 is applied, changing the CONSEQUENCES OF ERRORS
9137 section to “Default.”.

9138 **NAME**

9139 chmod — change the file modes

9140 **SYNOPSIS**

9141 chmod [-R] mode file ...

9142 **DESCRIPTION**9143 The *chmod* utility shall change any or all of the file mode bits of the file named by each *file* operand in the way specified by the *mode* operand.9145 It is implementation-defined whether and how the *chmod* utility affects any alternate or
9146 additional file access control mechanism (see the Base Definitions volume of
9147 IEEE Std 1003.1-2001, Section 4.4, File Access Permissions) being used for the specified file.9148 Only a process whose effective user ID matches the user ID of the file, or a process with the
9149 appropriate privileges, shall be permitted to change the file mode bits of a file.9150 **OPTIONS**9151 The *chmod* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
9152 12.2, Utility Syntax Guidelines.

9153 The following option shall be supported:

9154 **-R** Recursively change file mode bits. For each *file* operand that names a directory,
9155 *chmod* shall change the file mode bits of the directory and all files in the file
9156 hierarchy below it.9157 **OPERANDS**

9158 The following operands shall be supported:

9159 *mode* Represents the change to be made to the file mode bits of each file named by one of
9160 the *file* operands; see the EXTENDED DESCRIPTION section.9161 *file* A pathname of a file whose file mode bits shall be modified.9162 **STDIN**

9163 Not used.

9164 **INPUT FILES**

9165 None.

9166 **ENVIRONMENT VARIABLES**9167 The following environment variables shall affect the execution of *chmod*:9168 **LANG** Provide a default value for the internationalization variables that are unset or null.
9169 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
9170 Internationalization Variables for the precedence of internationalization variables
9171 used to determine the values of locale categories.)9172 **LC_ALL** If set to a non-empty string value, override the values of all the other
9173 internationalization variables.9174 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
9175 characters (for example, single-byte as opposed to multi-byte characters in
9176 arguments).9177 **LC_MESSAGES**9178 Determine the locale that should be used to affect the format and contents of
9179 diagnostic messages written to standard error.

9180 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

9181 **ASYNCHRONOUS EVENTS**

9182 Default.

9183 **STDOUT**

9184 Not used.

9185 **STDERR**

9186 The standard error shall be used only for diagnostic messages.

9187 **OUTPUT FILES**

9188 None.

9189 **EXTENDED DESCRIPTION**

9190 The *mode* operand shall be either a *symbolic_mode* expression or a non-negative octal integer. The
9191 *symbolic_mode* form is described by the grammar later in this section.

9192 Each **clause** shall specify an operation to be performed on the current file mode bits of each *file*.
9193 The operations shall be performed on each *file* in the order in which the **clauses** are specified.

9194 The **who** symbols **u**, **g**, and **o** shall specify the *user*, *group*, and *other* parts of the file mode bits,
9195 respectively. A **who** consisting of the symbol **a** shall be equivalent to **ugo**.

9196 The **perm** symbols **r**, **w**, and **x** represent the *read*, *write*, and *execute/search* portions of file mode
9197 bits, respectively. The **perm** symbol **s** shall represent the *set-user-ID-on-execution* (when **who**
9198 contains or implies **u**) and *set-group-ID-on-execution* (when **who** contains or implies **g**) bits.

9199 The **perm** symbol **X** shall represent the execute/search portion of the file mode bits if the file is a
9200 directory or if the current (unmodified) file mode bits have at least one of the execute bits
9201 (**S_IXUSR**, **S_IXGRP**, or **S_IXOTH**) set. It shall be ignored if the file is not a directory and none of
9202 the execute bits are set in the current file mode bits.

9203 The **permcopy** symbols **u**, **g**, and **o** shall represent the current permissions associated with the
9204 *user*, *group*, and *other* parts of the file mode bits, respectively. For the remainder of this section,
9205 **perm** refers to the non-terminals **perm** and **permcopy** in the grammar.

9206 If multiple **actionlists** are grouped with a single **wholist** in the grammar, each **actionlist** shall be
9207 applied in the order specified with that **wholist**. The *op* symbols shall represent the operation
9208 performed, as follows:

- 9209 + If **perm** is not specified, the '+' operation shall not change the file mode bits.

9210 If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and
9211 other permissions, except for those with corresponding bits in the file mode creation mask
9212 of the invoking process, shall be set.

9213 Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be set.

- 9214 - If **perm** is not specified, the '-' operation shall not change the file mode bits.

9215 If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and
9216 other permissions, except for those with corresponding bits in the file mode creation mask
9217 of the invoking process, shall be cleared.

9218 Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be
9219 cleared.

- 9220 = Clear the file mode bits specified by the **who** value, or, if no **who** value is specified, all of the
9221 file mode bits specified in this volume of IEEE Std 1003.1-2001.

9222 If **perm** is not specified, the '=' operation shall make no further modifications to the file
 9223 mode bits.

9224 If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and
 9225 other permissions, except for those with corresponding bits in the file mode creation mask
 9226 of the invoking process, shall be set.

9227 Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be set.

9228 When using the symbolic mode form on a regular file, it is implementation-defined whether or
 9229 not:

- 9230 • Requests to set the set-user-ID-on-execution or set-group-ID-on-execution bit when all
 9231 execute bits are currently clear and none are being set are ignored.
- 9232 • Requests to clear all execute bits also clear the set-user-ID-on-execution and set-group-ID-
 9233 on-execution bits.
- 9234 • Requests to clear the set-user-ID-on-execution or set-group-ID-on-execution bits when all
 9235 execute bits are currently clear are ignored. However, if the command *ls -l file* writes an *s* in
 9236 the position indicating that the set-user-ID-on-execution or set-group-ID-on-execution is set,
 9237 the commands *chmod u-s file* or *chmod g-s file*, respectively, shall not be ignored.

9238 When using the symbolic mode form on other file types, it is implementation-defined whether
 9239 or not requests to set or clear the set-user-ID-on-execution or set-group-ID-on-execution bits are
 9240 honored.

9241 If the **who** symbol **o** is used in conjunction with the **perm** symbol **s** with no other **who** symbols
 9242 being specified, the set-user-ID-on-execution and set-group-ID-on-execution bits shall not be
 9243 modified. It shall not be an error to specify the **who** symbol **o** in conjunction with the **perm**
 9244 symbol **s**.

9245 XSI The **perm** symbol **t** shall specify the S_ISVTX bit and shall apply to directories only. The effect
 9246 when using it with any other file type is unspecified. It can be used with the **who** symbols **o**, **a**,
 9247 or with no **who** symbol. It shall not be an error to specify a **who** symbol of **u** or **g** in conjunction
 9248 with the **perm** symbol **t**; it shall be ignored for **u** and **g**.

9249 For an octal integer *mode* operand, the file mode bits shall be set absolutely.

9250 For each bit set in the octal number, the corresponding file permission bit shown in the following
 9251 table shall be set; all other file permission bits shall be cleared. For regular files, for each bit set in
 9252 the octal number corresponding to the set-user-ID-on-execution or the set-group-ID-on-
 9253 execution, bits shown in the following table shall be set; if these bits are not set in the octal
 9254 number, they are cleared. For other file types, it is implementation-defined whether or not
 9255 requests to set or clear the set-user-ID-on-execution or set-group-ID-on-execution bits are
 9256 honored.

Octal	Mode Bit	Octal	Mode Bit	Octal	Mode Bit	Octal	Mode Bit
4000	S_ISUID	0400	S_IRUSR	0040	S_IRGRP	0004	S_IROTH
2000	S_ISGID	0200	S_IWUSR	0020	S_IWGRP	0002	S_IWOTH
XSI 1000	S_ISVTX	0100	S_IXUSR	0010	S_IXGRP	0001	S_IXOTH

9261 When bits are set in the octal number other than those listed in the table above, the behavior is
 9262 unspecified.

9263 **Grammar for chmod**

9264 The grammar and lexical conventions in this section describe the syntax for the *symbolic_mode*
9265 operand. The general conventions for this style of grammar are described in Section 1.10 (on
9266 page 19). A valid *symbolic_mode* can be represented as the non-terminal symbol *symbolic_mode* in
9267 the grammar. This formal syntax shall take precedence over the preceding text syntax
9268 description.

9269 The lexical processing is based entirely on single characters. Implementations need not allow
9270 <blank>s within the single argument being processed.

```
9271         %start     symbolic_mode
9272         %%
9273         symbolic_mode    : clause
9274                | symbolic_mode ',' clause
9275               ;
9276         clause       : actionlist
9277                | wholist actionlist
9278               ;
9279         wholist      : who
9280                | wholist who
9281               ;
9282         who          : 'u' | 'g' | 'o' | 'a'
9283               ;
9284         actionlist   : action
9285                | actionlist action
9286               ;
9287         action       : op
9288                | op permplist
9289                | op permcopy
9290               ;
9291         permcopy     : 'u' | 'g' | 'o'
9292               ;
9293         op           : '+' | '-' | '='
9294               ;
9295         permplist    : perm
9296                | perm permplist
9297               ;
9298 XSI        perm      : 'r' | 'w' | 'x' | 'X' | 's' | 't'
9299               ;
```

9300 **EXIT STATUS**

9301 The following exit values shall be returned:

- 9302 0 The utility executed successfully and all requested changes were made.
9303 >0 An error occurred.

9304 CONSEQUENCES OF ERRORS

9305 Default.

9306 APPLICATION USAGE

9307 Some implementations of the *chmod* utility change the mode of a directory before the files in the
 9308 directory when performing a recursive (-R option) change; others change the directory mode
 9309 after the files in the directory. If an application tries to remove read or search permission for a
 9310 file hierarchy, the removal attempt fails if the directory is changed first; on the other hand, trying
 9311 to re-enable permissions to a restricted hierarchy fails if directories are changed last. Users
 9312 should not try to make a hierarchy inaccessible to themselves.

9313 Some implementations of *chmod* never used the process' *umask* when changing modes; systems
 9314 conformant with this volume of IEEE Std 1003.1-2001 do so when **who** is not specified. Note the
 9315 difference between:

9316 `chmod a-w file`

9317 which removes all write permissions, and:

9318 `chmod -- -w file`

9319 which removes write permissions that would be allowed if **file** was created with the same
 9320 *umask*.

9321 Conforming applications should never assume that they know how the set-user-ID and set-
 9322 group-ID bits on directories are interpreted.

9323 EXAMPLES

Mode	Results
<i>a+=</i>	Equivalent to <i>a+,a=</i> ; clears all file mode bits.
<i>go+-w</i>	Equivalent to <i>go+,go-w</i> ; clears group and other write bits.
<i>g=o-w</i>	Equivalent to <i>g=o,g-w</i> ; sets group bit to match other bits and then clears group write bit.
<i>g-r+w</i>	Equivalent to <i>g-r,g+w</i> ; clears group read bit and sets group write bit.
<i>uo=g</i>	Sets owner bits to match group bits and sets other bits to match group bits.

9334 RATIONALE

9335 The functionality of *chmod* is described substantially through references to concepts defined in
 9336 the System Interfaces volume of IEEE Std 1003.1-2001. In this way, there is less duplication of
 9337 effort required for describing the interactions of permissions. However, the behavior of this
 9338 utility is not described in terms of the *chmod()* function from the System Interfaces volume of
 9339 IEEE Std 1003.1-2001 because that specification requires certain side effects upon alternate file
 9340 access control mechanisms that might not be appropriate, depending on the implementation.

9341 Implementations that support mandatory file and record locking as specified by the 1984
 9342 /usr/group standard historically used the combination of set-group-ID bit set and group
 9343 execute bit clear to indicate mandatory locking. This condition is usually set or cleared with the
 9344 symbolic mode **perm** symbol **l** instead of the **perm** symbols **s** and **x** so that the mandatory
 9345 locking mode is not changed without explicit indication that that was what the user intended.
 9346 Therefore, the details on how the implementation treats these conditions must be defined in the
 9347 documentation. This volume of IEEE Std 1003.1-2001 does not require mandatory locking (nor
 9348 does the System Interfaces volume of IEEE Std 1003.1-2001), but does allow it as an extension.
 9349 However, this volume of IEEE Std 1003.1-2001 does require that the *ls* and *chmod* utilities work

9350 consistently in this area. If *ls -l file* indicates that the set-group-ID bit is set, *chmod g-s file* must
9351 clear it (assuming appropriate privileges exist to change modes).

9352 The System V and BSD versions use different exit status codes. Some implementations used the
9353 exit status as a count of the number of errors that occurred; this practice is unworkable since it
9354 can overflow the range of valid exit status values. This problem is avoided here by specifying
9355 only 0 and >0 as exit values.

9356 The System Interfaces volume of IEEE Std 1003.1-2001 indicates that implementation-defined
9357 restrictions may cause the S_ISUID and S_ISGID bits to be ignored. This volume of
9358 IEEE Std 1003.1-2001 allows the *chmod* utility to choose to modify these bits before calling
9359 *chmod()* (or some function providing equivalent capabilities) for non-regular files. Among other
9360 things, this allows implementations that use the set-user-ID and set-group-ID bits on directories
9361 to enable extended features to handle these extensions in an intelligent manner.

9362 The **X perm** symbol was adopted from BSD-based systems because it provides commonly
9363 desired functionality when doing recursive (-R option) modifications. Similar functionality is
9364 not provided by the *find* utility. Historical BSD versions of *chmod*, however, only supported **X**
9365 with *op+*; it has been extended in this volume of IEEE Std 1003.1-2001 because it is also useful
9366 with *op-*. (It has also been added for *op-* even though it duplicates **x**, in this case, because it is
9367 intuitive and easier to explain.)

9368 The grammar was extended with the *permcopy* non-terminal to allow historical-practice forms of
9369 symbolic modes like **o=u -g** (that is, set the “other” permissions to the permissions of “owner”
9370 minus the permissions of “group”).

9371 FUTURE DIRECTIONS

9372 None.

9373 SEE ALSO

9374 *ls*, *umask*, the System Interfaces volume of IEEE Std 1003.1-2001, *chmod()*

9375 CHANGE HISTORY

9376 First released in Issue 2.

9377 Issue 6

9378 The following new requirements on POSIX implementations derive from alignment with the
9379 Single UNIX Specification:

- 9380 • Octal modes have been kept and made mandatory despite being marked obsolescent in the
9381 ISO POSIX-2:1993 standard.

9382 IEEE PASC Interpretation 1003.2 #172 is applied, changing the CONSEQUENCES OF ERRORS
9383 section to “Default.”.

9384 The Open Group Base Resolution bwg2001-010 is applied, adding the description of the
9385 S_ISVTX bit and the **t perm** symbol as an XSI extension.

9386 **NAME**9387 *chown* — change the file ownership9388 **SYNOPSIS**9389 *chown -hR owner[:group] file ...*9390 *chown -R [-H | -L | -P] owner[:group] file ...*9391 **DESCRIPTION**9392 The *chown* utility shall set the user ID of the file named by each *file* operand to the user ID specified by the *owner* operand.9394 For each *file* operand, or, if the **-R** option is used, each file encountered while walking the
9395 directory trees specified by the *file* operands, the *chown* utility shall perform actions equivalent to
9396 the *chown()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001, called
9397 with the following arguments:

- 9398 1. The *file* operand shall be used as the *path* argument.
- 9399 2. The user ID indicated by the *owner* portion of the first operand shall be used as the *owner* argument.
- 9400 3. If the *group* portion of the first operand is given, the group ID indicated by it shall be used as the *group* argument; otherwise, the group ownership shall not be changed.

9403 Unless *chown* is invoked by a process with appropriate privileges, the set-user-ID and set-group-ID bits of a regular file shall be cleared upon successful completion; the set-user-ID and set-group-ID bits of other file types may be cleared.9406 **OPTIONS**9407 The *chown* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
9408 12.2, Utility Syntax Guidelines.

9409 The following options shall be supported by the implementation:

- | | |
|----------------|---|
| 9410 -h | If the system supports user IDs for symbolic links, for each <i>file</i> operand that names
9411 a file of type symbolic link, <i>chown</i> shall attempt to set the user ID of the symbolic
9412 link. If the system supports group IDs for symbolic links, and a group ID was
9413 specified, for each <i>file</i> operand that names a file of type symbolic link, <i>chown</i> shall
9414 attempt to set the group ID of the symbolic link. If the system does not support
9415 user or group IDs for symbolic links, for each <i>file</i> operand that names a file of type
9416 symbolic link, <i>chown</i> shall do nothing more with the current file and shall go on to
9417 any remaining files. |
| 9418 -H | If the -R option is specified and a symbolic link referencing a file of type directory
9419 is specified on the command line, <i>chown</i> shall change the user ID (and group ID, if
9420 specified) of the directory referenced by the symbolic link and all files in the file
9421 hierarchy below it. |
| 9422 -L | If the -R option is specified and a symbolic link referencing a file of type directory
9423 is specified on the command line or encountered during the traversal of a file
9424 hierarchy, <i>chown</i> shall change the user ID (and group ID, if specified) of the
9425 directory referenced by the symbolic link and all files in the file hierarchy below it. |
| 9426 -P | If the -R option is specified and a symbolic link is specified on the command line
9427 or encountered during the traversal of a file hierarchy, <i>chown</i> shall change the
9428 owner ID (and group ID, if specified) of the symbolic link if the system supports
9429 this operation. The <i>chown</i> utility shall not follow the symbolic link to any other
9430 part of the file hierarchy. |

9431 **-R** Recursively change file user and group IDs. For each *file* operand that names a
9432 directory, *chown* shall change the user ID (and group ID, if specified) of the
9433 directory and all files in the file hierarchy below it. Unless a **-H**, **-L**, or **-P** option is
9434 specified, it is unspecified which of these options will be used as the default.

9435 Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be
9436 considered an error. The last option specified shall determine the behavior of the utility.

9437 OPERANDS

9438 The following operands shall be supported:

9439 *owner[:group]* A user ID and optional group ID to be assigned to *file*. The *owner* portion of this
9440 operand shall be a user name from the user database or a numeric user ID. Either
9441 specifies a user ID which shall be given to each file named by one of the *file*
9442 operands. If a numeric *owner* operand exists in the user database as a user name,
9443 the user ID number associated with that user name shall be used as the user ID.
9444 Similarly, if the *group* portion of this operand is present, it shall be a group name
9445 from the group database or a numeric group ID. Either specifies a group ID which
9446 shall be given to each file. If a numeric group operand exists in the group database
9447 as a group name, the group ID number associated with that group name shall be
9448 used as the group ID.

9449 *file* A pathname of a file whose user ID is to be modified.

9450 STDIN

9451 Not used.

9452 INPUT FILES

9453 None.

9454 ENVIRONMENT VARIABLES

9455 The following environment variables shall affect the execution of *chown*:

9456 **LANG** Provide a default value for the internationalization variables that are unset or null.
9457 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
9458 Internationalization Variables for the precedence of internationalization variables
9459 used to determine the values of locale categories.)

9460 **LC_ALL** If set to a non-empty string value, override the values of all the other
9461 internationalization variables.

9462 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
9463 characters (for example, single-byte as opposed to multi-byte characters in
9464 arguments).

9465 **LC_MESSAGES**

9466 Determine the locale that should be used to affect the format and contents of
9467 diagnostic messages written to standard error.

9468 **XSI NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

9469 ASYNCHRONOUS EVENTS

9470 Default.

9471 STDOUT

9472 Not used.

9473 STDRERR

9474 The standard error shall be used only for diagnostic messages.

9475 OUTPUT FILES

9476 None.

9477 EXTENDED DESCRIPTION

9478 None.

9479 EXIT STATUS

9480 The following exit values shall be returned:

9481 0 The utility executed successfully and all requested changes were made.

9482 >0 An error occurred.

9483 CONSEQUENCES OF ERRORS

9484 Default.

9485 APPLICATION USAGE

9486 Only the owner of a file or the user with appropriate privileges may change the owner or group of a file.

9488 Some implementations restrict the use of *chown* to a user with appropriate privileges.

9489 EXAMPLES

9490 None.

9491 RATIONALE

9492 The System V and BSD versions use different exit status codes. Some implementations used the exit status as a count of the number of errors that occurred; this practice is unworkable since it can overflow the range of valid exit status values. These are masked by specifying only 0 and >0 as exit values.

9496 The functionality of *chown* is described substantially through references to functions in the System Interfaces volume of IEEE Std 1003.1-2001. In this way, there is no duplication of effort required for describing the interactions of permissions, multiple groups, and so on.

9499 The 4.3 BSD method of specifying both owner and group was included in this volume of IEEE Std 1003.1-2001 because:

- 9501 • There are cases where the desired end condition could not be achieved using the *chgrp* and
9502 *chown* (that only changed the user ID) utilities. (If the current owner is not a member of the
9503 desired group and the desired owner is not a member of the current group, the *chown()*
9504 function could fail unless both owner and group are changed at the same time.)
- 9505 • Even if they could be changed independently, in cases where both are being changed, there is
9506 a 100% performance penalty caused by being forced to invoke both utilities.

9507 The BSD syntax *user[group]* was changed to *user[:group]* in this volume of IEEE Std 1003.1-2001
9508 because the period is a valid character in login names (as specified by the Base Definitions
9509 volume of IEEE Std 1003.1-2001, login names consist of characters in the portable filename
9510 character set). The colon character was chosen as the replacement for the period character
9511 because it would never be allowed as a character in a user name or group name on historical
9512 implementations.

9513 The **-R** option is considered by some observers as an undesirable departure from the historical
9514 UNIX system tools approach; since a tool, *find*, already exists to recurse over directories, there
9515 seemed to be no good reason to require other tools to have to duplicate that functionality.
9516 However, the **-R** option was deemed an important user convenience, is far more efficient than

9517 forking a separate process for each element of the directory hierarchy, and is in widespread
9518 historical use.

9519 **FUTURE DIRECTIONS**

9520 None.

9521 **SEE ALSO**

9522 *chmod*, *chgrp*, the System Interfaces volume of IEEE Std 1003.1-2001, *chown()*

9523 **CHANGE HISTORY**

9524 First released in Issue 2.

9525 **Issue 6**

9526 New options **-h**, **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These
9527 options affect the processing of symbolic links.

9528 The normative text is reworded to avoid use of the term “must” for application requirements.

9529 IEEE PASC Interpretation 1003.2 #172 is applied, changing the CONSEQUENCES OF ERRORS
9530 section to “Default.”.

9531 The “otherwise, …” text in item 3. of the DESCRIPTION is changed to “otherwise, the group
9532 ownership shall not be changed”.

9533 NAME

9534 cksum — write file checksums and sizes

9535 SYNOPSIS

9536 cksum [*file* ...]

9537 DESCRIPTION

9538 The *cksum* utility shall calculate and write to standard output a cyclic redundancy check (CRC)
9539 for each input file, and also write to standard output the number of octets in each file. The CRC
9540 used is based on the polynomial used for CRC error checking in the ISO/IEC 8802-3:1996
9541 standard (Ethernet).

9542 The encoding for the CRC checksum is defined by the generating polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

9544 Mathematically, the CRC value corresponding to a given file shall be defined by the following
9545 procedure:

- 9546 1. The *n* bits to be evaluated are considered to be the coefficients of a mod 2 polynomial *M(x)*
9547 of degree *n*-1. These *n* bits are the bits from the file, with the most significant bit being the
9548 most significant bit of the first octet of the file and the last bit being the least significant bit
9549 of the last octet, padded with zero bits (if necessary) to achieve an integral number of
9550 octets, followed by one or more octets representing the length of the file as a binary value,
9551 least significant octet first. The smallest number of octets capable of representing this
9552 integer shall be used.
- 9553 2. *M(x)* is multiplied by x^{32} (that is, shifted left 32 bits) and divided by *G(x)* using mod 2
9554 division, producing a remainder *R(x)* of degree ≤ 31 .
- 9555 3. The coefficients of *R(x)* are considered to be a 32-bit sequence.
- 9556 4. The bit sequence is complemented and the result is the CRC.

9557 OPTIONS

9558 None.

9559 OPERANDS

9560 The following operand shall be supported:

9561 *file* A pathname of a file to be checked. If no *file* operands are specified, the standard
9562 input shall be used.

9563 STDIN

9564 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES
9565 section.

9566 INPUT FILES

9567 The input files can be any file type.

9568 ENVIRONMENT VARIABLES

9569 The following environment variables shall affect the execution of *cksum*:

9570 *LANG* Provide a default value for the internationalization variables that are unset or null.
9571 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
9572 Internationalization Variables for the precedence of internationalization variables
9573 used to determine the values of locale categories.)

9574 *LC_ALL* If set to a non-empty string value, override the values of all the other
9575 internationalization variables.

9576	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
9579	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
9582	XSI <i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
9583	ASYNCHRONOUS EVENTS	
9584		Default.
9585	STDOUT	
9586		For each file processed successfully, the <i>cksum</i> utility shall write in the following format:
9587		"%u %d %s\n", <checksum>, <# of octets>, <pathname>
9588		If no <i>file</i> operand was specified, the pathname and its leading <space> shall be omitted.
9589	STDERR	
9590		The standard error shall be used only for diagnostic messages.
9591	OUTPUT FILES	
9592		None.
9593	EXTENDED DESCRIPTION	
9594		None.
9595	EXIT STATUS	
9596		The following exit values shall be returned:
9597		0 All files were processed successfully.
9598		>0 An error occurred.
9599	CONSEQUENCES OF ERRORS	
9600		Default.
9601	APPLICATION USAGE	
9602		The <i>cksum</i> utility is typically used to quickly compare a suspect file against a trusted version of the same, such as to ensure that files transmitted over noisy media arrive intact. However, this comparison cannot be considered cryptographically secure. The chances of a damaged file producing the same CRC as the original are small; deliberate deception is difficult, but probably not impossible.
9607		Although input files to <i>cksum</i> can be any type, the results need not be what would be expected on character special device files or on file types not described by the System Interfaces volume of IEEE Std 1003.1-2001. Since this volume of IEEE Std 1003.1-2001 does not specify the block size used when doing input, checksums of character special files need not process all of the data in those files.
9612		The algorithm is expressed in terms of a bitstream divided into octets. If a file is transmitted between two systems and undergoes any data transformation (such as changing little-endian byte ordering to big-endian), identical CRC values cannot be expected. Implementations performing such transformations may extend <i>cksum</i> to handle such situations.

9616 EXAMPLES

9617 None.

9618 RATIONALE

9619 The following C-language program can be used as a model to describe the algorithm. It assumes
 9620 that a **char** is one octet. It also assumes that the entire file is available for one pass through the
 9621 function. This was done for simplicity in demonstrating the algorithm, rather than as an
 9622 implementation model.

```
9623 static unsigned long crctab[] = {
9624     0x00000000,
9625     0x04c11db7, 0x09823b6e, 0x0d4326d9, 0x130476dc, 0x17c56b6b,
9626     0x1a864db2, 0x1e475005, 0x2608edb8, 0x22c9f00f, 0x2f8ad6d6,
9627     0x2b4bcb61, 0x350c9b64, 0x31cd86d3, 0x3c8ea00a, 0x384fbdbd,
9628     0x4c11db70, 0x48d0c6c7, 0x4593e01e, 0x4152fd9, 0x5f15adac,
9629     0x5bd4b01b, 0x569796c2, 0x52568b75, 0x6a1936c8, 0x6ed82b7f,
9630     0x639b0da6, 0x675a1011, 0x791d4014, 0x7ddc5da3, 0x709f7b7a,
9631     0x745e66cd, 0x9823b6e0, 0x9ce2ab57, 0x91a18d8e, 0x95609039,
9632     0x8b27c03c, 0x8fe6dd8b, 0x82a5fb52, 0x8664e6e5, 0xbe2b5b58,
9633     0xbaea46ef, 0xb7a96036, 0xb3687d81, 0xad2f2d84, 0xa9ee3033,
9634     0xa4ad16ea, 0xa06c0b5d, 0xd4326d90, 0xd0f37027, 0xddb056fe,
9635     0xd9714b49, 0xc7361b4c, 0xc3f706fb, 0xceb42022, 0xca753d95,
9636     0xf23a8028, 0xf6fb9d9f, 0xfbb8bb46, 0xff79a6f1, 0xe13ef6f4,
9637     0xe5ffeb43, 0xe8bccd9a, 0xec7ddd02d, 0x34867077, 0x30476dc0,
9638     0x3d044b19, 0x39c556ae, 0x278206ab, 0x23431b1c, 0x2e003dc5,
9639     0x2ac12072, 0x128e9dcf, 0x164f8078, 0x1b0ca6a1, 0x1fcdbb16,
9640     0x018aeb13, 0x054bf6a4, 0x0808d07d, 0x0cc9cdca, 0x7897ab07,
9641     0x7c56b6b0, 0x71159069, 0x75d48dde, 0x6b93dddb, 0x6f52c06c,
9642     0x6211e6b5, 0x66d0fb02, 0x5e9f46bf, 0x5a5e5b08, 0x571d7dd1,
9643     0x53dc6066, 0x4d9b3063, 0x495a2dd4, 0x44190b0d, 0x40d816ba,
9644     0xacaca5c697, 0xa864db20, 0xa527fdf9, 0xa1e6e04e, 0xbfa1b04b,
9645     0xbb60adfc, 0xb6238b25, 0xb2e29692, 0x8aad2b2f, 0x8e6c3698,
9646     0x832f1041, 0x87ee0df6, 0x99a95df3, 0x9d684044, 0x902b669d,
9647     0x94ea7b2a, 0xe0b41de7, 0xe4750050, 0xe9362689, 0xedf73b3e,
9648     0xf3b06b3b, 0xf771768c, 0xfa325055, 0xefef34de2, 0xc6bcf05f,
9649     0xc27dede8, 0xcf3ecb31, 0xcbffd686, 0xd5b88683, 0xd1799b34,
9650     0xdc3abded, 0xd8fba05a, 0x690ce0ee, 0x6dcdf59, 0x608edb80,
9651     0x644fc637, 0x7a089632, 0x7ec98b85, 0x738aad5c, 0x774bb0eb,
9652     0x4f040d56, 0x4bc510e1, 0x46863638, 0x42472b8f, 0x5c007b8a,
9653     0x58c1663d, 0x558240e4, 0x51435d53, 0x251d3b9e, 0x21dc2629,
9654     0x2c9f00f0, 0x285e1d47, 0x36194d42, 0x32d850f5, 0x3f9b762c,
9655     0x3b5a6b9b, 0x0315d626, 0x07d4cb91, 0xa97ed48, 0xe56f0ff,
9656     0x1011a0fa, 0x14d0bd4d, 0x19939b94, 0x1d528623, 0xf12f560e,
9657     0xf5ee4bb9, 0xf8ad6d60, 0xfc6c70d7, 0xe22b20d2, 0xe6ea3d65,
9658     0xeba91bbc, 0xef68060b, 0xd727bbb6, 0xd3e6a601, 0xdeaa580d8,
9659     0xda649d6f, 0xc423cd6a, 0xc0e2d0dd, 0xcdaf604, 0xc960ebb3,
9660     0xbd3e8d7e, 0xb9ff90c9, 0xb4bcb610, 0xb07daba7, 0xae3afba2,
9661     0xaafbe615, 0xa7b8c0cc, 0xa379dd7b, 0xb3660c6, 0x9ff77d71,
9662     0x92b45ba8, 0x9675461f, 0x8832161a, 0x8cf30bad, 0x81b02d74,
9663     0x857130c3, 0x5d8a9099, 0x594b8d2e, 0x5408abf7, 0x50c9b640,
9664     0x4e8ee645, 0x4a4ffbf2, 0x470cdd2b, 0x43cdc09c, 0x7b827d21,
9665     0x7f436096, 0x7200464f, 0x76c15bf8, 0x68860bfd, 0x6c47164a,
9666     0x61043093, 0x65c52d24, 0x119b4be9, 0x155a565e, 0x18197087,
```

```

9667      0x1cd86d30, 0x029f3d35, 0x065e2082, 0x0b1d065b, 0x0fdc1bec,
9668      0x3793a651, 0x3352bbe6, 0x3e119d3f, 0x3ad08088, 0x2497d08d,
9669      0x2056cd3a, 0x2d15ebe3, 0x29d4f654, 0xc5a92679, 0xc1683bce,
9670      0xcc2b1d17, 0xc8ea00a0, 0xd6ad50a5, 0xd26c4d12, 0xdf2f6bcb,
9671      0xdbbe767c, 0xe3a1cbc1, 0xe760d676, 0xea23f0af, 0xeee2ed18,
9672      0xf0a5bd1d, 0xf464a0aa, 0xf9278673, 0xfde69bc4, 0x89b8fd09,
9673      0x8d79e0be, 0x803ac667, 0x84fbdbd0, 0x9abc8bd5, 0x9e7d9662,
9674      0x933eb0bb, 0x97ffad0c, 0xafb010b1, 0xab710d06, 0xa6322bdf,
9675      0xa2f33668, 0xbcb4666d, 0xb8757bda, 0xb5365d03, 0xb1f740b4
9676  };

9677  unsigned long memcrc(const unsigned char *b, size_t n)
9678  {
9679  /* Input arguments:
9680   * const char* b == byte sequence to checksum
9681   * size_t n == length of sequence
9682   */
9683      register unsigned i, c, s = 0;
9684      for (i = n; i > 0; --i) {
9685          c = (unsigned)(*b++);
9686          s = (s << 8) ^ crctab[(s >> 24) ^ c];
9687      }
9688      /* Extend with the length of the string. */
9689      while (n != 0) {
9690          c = n & 0377;
9691          n >>= 8;
9692          s = (s << 8) ^ crctab[(s >> 24) ^ c];
9693      }
9694      return ~s;
9695  }

```

9696 The historical practice of writing the number of “blocks” has been changed to writing the
9697 number of octets, since the latter is not only more useful, but also since historical
9698 implementations have not been consistent in defining what a “block” meant. Octets are used
9699 instead of bytes because bytes can differ in size between systems.

9700 The algorithm used was selected to increase the operational robustness of *cksum*. Neither the
9701 System V nor BSD *sum* algorithm was selected. Since each of these was different and each was
9702 the default behavior on those systems, no realistic compromise was available if either were
9703 selected—some set of historical applications would break. Therefore, the name was changed to
9704 *cksum*. Although the historical *sum* commands will probably continue to be provided for many
9705 years, programs designed for portability across systems should use the new name.

9706 The algorithm selected is based on that used by the ISO/IEC 8802-3:1996 standard (Ethernet) for
9707 the frame check sequence field. The algorithm used does not match the technical definition of a
9708 *checksum*; the term is used for historical reasons. The length of the file is included in the CRC
9709 calculation because this parallels inclusion of a length field by Ethernet in its CRC, but also
9710 because it guards against inadvertent collisions between files that begin with different series of
9711 zero octets. The chance that two different files produce identical CRCs is much greater when
9712 their lengths are not considered. Keeping the length and the checksum of the file itself separate
9713 would yield a slightly more robust algorithm, but historical usage has always been that a single
9714 number (the checksum as printed) represents the signature of the file. It was decided that

9715 historical usage was the more important consideration.

9716 Early proposals contained modifications to the Ethernet algorithm that involved extracting table
9717 values whenever an intermediate result became zero. This was demonstrated to be less robust
9718 than the current method and mathematically difficult to describe or justify.

9719 The calculation used is identical to that given in pseudo-code in the referenced Sarwate article.
9720 The pseudo-code rendition is:

```
9721 X <- 0; Y <- 0;
9722 for i <- m -1 step -1 until 0 do
9723 begin
9724   T <- X(1) ^ A[i];
9725   X(1) <- X(0); X(0) <- Y(1); Y(1) <- Y(0); Y(0) <- 0;
9726   comment: f[T] and f'[T] denote the T-th words in the
9727     table f and f' ;
9728   X <- X ^ f[T]; Y <- Y ^ f'[T];
9729 end
```

9730 The pseudo-code is reproduced exactly as given; however, note that in the case of **cksum**, **A[i]**
9731 represents a byte of the file, the words **X** and **Y** are treated as a single 32-bit value, and the tables
9732 **f** and **f'** are a single table containing 32-bit values.

9733 The referenced Sarwate article also discusses generating the table.

9734 FUTURE DIRECTIONS

9735 None.

9736 SEE ALSO

9737 None.

9738 CHANGE HISTORY

9739 First released in Issue 4.

NAME

9740 cmp — compare two files

SYNOPSIS

9743 `cmp [-l | -s] file1 file2`

DESCRIPTION

9745 The *cmp* utility shall compare two files. The *cmp* utility shall write no output if the files are the same. Under default options, if they differ, it shall write to standard output the byte and line number at which the first difference occurred. Bytes and lines shall be numbered beginning with 1.

OPTIONS

9750 The *cmp* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 9751 12.2, Utility Syntax Guidelines.

9752 The following options shall be supported:

9753 **-l** (Lowercase ell.) Write the byte number (decimal) and the differing bytes (octal) for 9754 each difference.

9755 **-s** Write nothing for differing files; return exit status only.

OPERANDS

9757 The following operands shall be supported:

9758 *file1* A pathname of the first file to be compared. If *file1* is ‘-’, the standard input shall 9759 be used.

9760 *file2* A pathname of the second file to be compared. If *file2* is ‘-’, the standard input 9761 shall be used.

9762 If both *file1* and *file2* refer to standard input or refer to the same FIFO special, block special, or 9763 character special file, the results are undefined.

STDIN

9765 The standard input shall be used only if the *file1* or *file2* operand refers to standard input. See the 9766 INPUT FILES section.

INPUT FILES

9768 The input files can be any file type.

ENVIRONMENT VARIABLES

9770 The following environment variables shall affect the execution of *cmp*:

9771 **LANG** Provide a default value for the internationalization variables that are unset or null. 9772 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, 9773 Internationalization Variables for the precedence of internationalization variables 9774 used to determine the values of locale categories.)

9775 **LC_ALL** If set to a non-empty string value, override the values of all the other 9776 internationalization variables.

9777 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as 9778 characters (for example, single-byte as opposed to multi-byte characters in 9779 arguments).

LC_MESSAGES

9781 Determine the locale that should be used to affect the format and contents of 9782 diagnostic messages written to standard error and informative messages written to 9783 standard output.

9784 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

9785 **ASYNCHRONOUS EVENTS**

9786 Default.

9787 **STDOUT**

9788 In the POSIX locale, results of the comparison shall be written to standard output. When no
9789 options are used, the format shall be:

9790 "%s %s differ: char %d, line %d\n", *file1*, *file2*,
9791 <byte number>, <line number>

9792 When the **-l** option is used, the format shall be:

9793 "%d %o %o\n", <byte number>, <differing byte>,
9794 <differing byte>

9795 for each byte that differs. The first <differing byte> number is from *file1* while the second is from
9796 *file2*. In both cases, <byte number> shall be relative to the beginning of the file, beginning with 1.

9797 No output shall be written to standard output when the **-s** option is used.

9798 **STDERR**

9799 The standard error shall be used only for diagnostic messages. If *file1* and *file2* are identical for
9800 the entire length of the shorter file, in the POSIX locale the following diagnostic message shall be
9801 written, unless the **-s** option is specified:

9802 "cmp: EOF on %s%s\n", <name of shorter file>, <additional info>

9803 The <additional info> field shall either be null or a string that starts with a <blank> and contains
9804 no <newline>s. Some implementations report on the number of lines in this case.

9805 **OUTPUT FILES**

9806 None.

9807 **EXTENDED DESCRIPTION**

9808 None.

9809 **EXIT STATUS**

9810 The following exit values shall be returned:

9811 0 The files are identical.

9812 1 The files are different; this includes the case where one file is identical to the first part of the
9813 other.

9814 >1 An error occurred.

9815 **CONSEQUENCES OF ERRORS**

9816 Default.

9817 **APPLICATION USAGE**

9818 Although input files to *cmp* can be any type, the results might not be what would be expected on
9819 character special device files or on file types not described by the System Interfaces volume of
9820 IEEE Std 1003.1-2001. Since this volume of IEEE Std 1003.1-2001 does not specify the block size
9821 used when doing input, comparisons of character special files need not compare all of the data
9822 in those files.

9823 For files which are not text files, line numbers simply reflect the presence of a <newline>,
9824 without any implication that the file is organized into lines.

9825 **EXAMPLES**

9826 None.

9827 **RATIONALE**

9828 The global language in Section 1.11 (on page 20) indicates that using two mutually-exclusive
9829 options together produces unspecified results. Some System V implementations consider the
9830 option usage:

9831 `cmp -l -s ...`

9832 to be an error. They also treat:

9833 `cmp -s -l ...`

9834 as if no options were specified. Both of these behaviors are considered bugs, but are allowed.

9835 The word **char** in the standard output format comes from historical usage, even though it is
9836 actually a byte number. When *cmp* is supported in other locales, implementations are
9837 encouraged to use the word *byte* or its equivalent in another language. Users should not
9838 interpret this difference to indicate that the functionality of the utility changed between locales.

9839 Some implementations report on the number of lines in the identical-but-shorter file case. This is
9840 allowed by the inclusion of the <additional info> fields in the output format. The restriction on
9841 having a leading <blank> and no <newline>s is to make parsing for the filename easier. It is
9842 recognized that some filenames containing white-space characters make parsing difficult
9843 anyway, but the restriction does aid programs used on systems where the names are
9844 predominantly well behaved.

9845 **FUTURE DIRECTIONS**

9846 None.

9847 **SEE ALSO**9848 *comm, diff*9849 **CHANGE HISTORY**

9850 First released in Issue 2.

9851 NAME

9852 comm — select or reject lines common to two files

9853 SYNOPSIS

9854 comm [-123] *file1 file2*

9855 DESCRIPTION

9856 The *comm* utility shall read *file1* and *file2*, which should be ordered in the current collating sequence, and produce three text columns as output: lines only in *file1*, lines only in *file2*, and lines in both files.

9859 If the lines in both files are not ordered according to the collating sequence of the current locale, the results are unspecified.

9861 OPTIONS

9862 The *comm* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

9864 The following options shall be supported:

9865 **-1** Suppress the output column of lines unique to *file1*.

9866 **-2** Suppress the output column of lines unique to *file2*.

9867 **-3** Suppress the output column of lines duplicated in *file1* and *file2*.

9868 OPERANDS

9869 The following operands shall be supported:

9870 *file1* A pathname of the first file to be compared. If *file1* is ‘-’, the standard input shall be used.

9872 *file2* A pathname of the second file to be compared. If *file2* is ‘-’, the standard input shall be used.

9874 If both *file1* and *file2* refer to standard input or to the same FIFO special, block special, or character special file, the results are undefined.

9876 STDIN

9877 The standard input shall be used only if one of the *file1* or *file2* operands refers to standard input.
9878 See the INPUT FILES section.

9879 INPUT FILES

9880 The input files shall be text files.

9881 ENVIRONMENT VARIABLES

9882 The following environment variables shall affect the execution of *comm*:

9883 *LANG* Provide a default value for the internationalization variables that are unset or null.
9884 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
9885 Internationalization Variables for the precedence of internationalization variables
9886 used to determine the values of locale categories.)

9887 *LC_ALL* If set to a non-empty string value, override the values of all the other
9888 internationalization variables.

9889 *LC_COLLATE*

9890 Determine the locale for the collating sequence *comm* expects to have been used
9891 when the input files were sorted.

9892 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
9893 characters (for example, single-byte as opposed to multi-byte characters in

9894 arguments and input files).

9895 ***LC_MESSAGES***
9896 Determine the locale that should be used to affect the format and contents of
9897 diagnostic messages written to standard error.

9898 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

9899 **ASYNCHRONOUS EVENTS**

9900 Default.

9901 **STDOUT**

9902 The *comm* utility shall produce output depending on the options selected. If the **-1**, **-2**, and **-3** options are all selected, *comm* shall write nothing to standard output.

9903 If the **-1** option is not selected, lines contained only in *file1* shall be written using the format:
9904 "*%s\n*", *<line in file1>*

9905 If the **-2** option is not selected, lines contained only in *file2* are written using the format:
9906 "*%s%s\n*", *<lead>, <line in file2>*

9907 where the string *<lead>* is as follows:

9908 *<tab>* The **-1** option is not selected.

9909 null string The **-1** option is selected.

9910 If the **-3** option is not selected, lines contained in both files shall be written using the format:
9911 "*%s%s\n*", *<lead>, <line in both>*

9912 where the string *<lead>* is as follows:

9913 *<tab><tab>* Neither the **-1** nor the **-2** option is selected.

9914 *<tab>* Exactly one of the **-1** and **-2** options is selected.

9915 null string Both the **-1** and **-2** options are selected.

9916 If the input files were ordered according to the collating sequence of the current locale, the lines written shall be in the collating sequence of the original lines.

9917

9918 **STDERR**

9919 The standard error shall be used only for diagnostic messages.

9920

9921 **OUTPUT FILES**

9922 None.

9923 **EXTENDED DESCRIPTION**

9924 None.

9925 **EXIT STATUS**

9926 The following exit values shall be returned:

9927 0 All input files were successfully output as specified.

9928 >0 An error occurred.

9929 **CONSEQUENCES OF ERRORS**

9930 Default.

9931 **APPLICATION USAGE**

9932 If the input files are not properly presorted, the output of *comm* might not be useful.

9933 **EXAMPLES**

9934 If a file named **xcu** contains a sorted list of the utilities in this volume of IEEE Std 1003.1-2001, a
9935 file named **xpg3** contains a sorted list of the utilities specified in the X/Open Portability Guide,
9936 Issue 3, and a file named **svid89** contains a sorted list of the utilities in the System V Interface
9937 Definition Third Edition:

9938 `comm -23 xcu xpg3 | comm -23 - svid89`

9939 would print a list of utilities in this volume of IEEE Std 1003.1-2001 not specified by either of the
9940 other documents:

9941 `comm -12 xcu xpg3 | comm -12 - svid89`

9942 would print a list of utilities specified by all three documents, and:

9943 `comm -12 xpg3 svid89 | comm -23 - xcu`

9944 would print a list of utilities specified by both XPG3 and the SVID, but not specified in this
9945 volume of IEEE Std 1003.1-2001.

9946 **RATIONALE**

9947 None.

9948 **FUTURE DIRECTIONS**

9949 None.

9950 **SEE ALSO**

9951 *cmp, diff, sort, uniq*

9952 **CHANGE HISTORY**

9953 First released in Issue 2.

9954 **Issue 6**

9955 The normative text is reworded to avoid use of the term “must” for application requirements.

9956 **NAME**

9957 command — execute a simple command

9958 **SYNOPSIS**9959 command [-p] *command_name* [*argument ...*]9960 UP command [-v | -V] *command_name*

9961

9962 **DESCRIPTION**

9963 The *command* utility shall cause the shell to treat the arguments as a simple command,
9964 suppressing the shell function lookup that is described in Section 2.9.1.1 (on page 48), item 1b.

9965 If the *command_name* is the same as the name of one of the special built-in utilities, the special
9966 properties in the enumerated list at the beginning of Section 2.14 (on page 64) shall not occur. In
9967 every other respect, if *command_name* is not the name of a function, the effect of *command* (with
9968 no options) shall be the same as omitting *command*.

9969 On systems supporting the User Portability Utilities option, the *command* utility also shall
9970 provide information concerning how a command name is interpreted by the shell; see **-v** and
9971 **-V**.

9972 **OPTIONS**

9973 The *command* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001,
9974 Section 12.2, Utility Syntax Guidelines.

9975 The following options shall be supported:

9976 **-p** Perform the command search using a default value for *PATH* that is guaranteed to
9977 find all of the standard utilities.

9978 **-v** (On systems supporting the User Portability Utilities option.) Write a string to
9979 standard output that indicates the pathname or command that will be used by the
9980 shell, in the current shell execution environment (see Section 2.12 (on page 61)), to
9981 invoke *command_name*, but do not invoke *command_name*.

- 9982 • Utilities, regular built-in utilities, *command_names* including a slash character,
9983 and any implementation-defined functions that are found using the *PATH*
9984 variable (as described in Section 2.9.1.1 (on page 48)), shall be written as
9985 absolute pathnames.
- 9986 • Shell functions, special built-in utilities, regular built-in utilities not associated
9987 with a *PATH* search, and shell reserved words shall be written as just their
9988 names.
- 9989 • An alias shall be written as a command line that represents its alias definition.
- 9990 • Otherwise, no output shall be written and the exit status shall reflect that the
9991 name was not found.

9992 **-V** (On systems supporting the User Portability Utilities option.) Write a string to
9993 standard output that indicates how the name given in the *command_name* operand
9994 will be interpreted by the shell, in the current shell execution environment (see
9995 Section 2.12 (on page 61)), but do not invoke *command_name*. Although the format
9996 of this string is unspecified, it shall indicate in which of the following categories
9997 *command_name* falls and shall include the information stated:

- 9998 • Utilities, regular built-in utilities, and any implementation-defined functions
9999 that are found using the *PATH* variable (as described in Section 2.9.1.1 (on page
10000 48)), shall be identified as such and include the absolute pathname in the string.

- 10001 • Other shell functions shall be identified as functions.
- 10002 • Aliases shall be identified as aliases and their definitions included in the string.
- 10003 • Special built-in utilities shall be identified as special built-in utilities.
- 10004 • Regular built-in utilities not associated with a *PATH* search shall be identified as regular built-in utilities. (The term “regular” need not be used.)
- 10005
- 10006 • Shell reserved words shall be identified as reserved words.

10007 OPERANDS

- 10008 The following operands shall be supported:
- 10009 *argument* One of the strings treated as an argument to *command_name*.
- 10010 *command_name*
- 10011 The name of a utility or a special built-in utility.

10012 STDIN

- 10013 Not used.

10014 INPUT FILES

- 10015 None.

10016 ENVIRONMENT VARIABLES

- 10017 The following environment variables shall affect the execution of *command*:

- 10018 *LANG* Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
Internationalization Variables for the precedence of internationalization variables
used to determine the values of locale categories.)
- 10022 *LC_ALL* If set to a non-empty string value, override the values of all the other
internationalization variables.
- 10024 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
characters (for example, single-byte as opposed to multi-byte characters in
arguments).
- 10027 *LC_MESSAGES*
10028 Determine the locale that should be used to affect the format and contents of
10029 diagnostic messages written to standard error and informative messages written to
10030 standard output.
- 10031 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.
- 10032 *PATH* Determine the search path used during the command search described in Section
10033 2.9.1.1 (on page 48), except as described under the **-p** option.

10034 ASYNCHRONOUS EVENTS

- 10035 Default.

10036 STDOUT

- 10037 When the **-v** option is specified, standard output shall be formatted as:

10038 "*%s\n*", <*pathname or command*>

- 10039 When the **-V** option is specified, standard output shall be formatted as:

10040 "*%s\n*", <*unspecified*>

10041 STDRERR

10042 The standard error shall be used only for diagnostic messages.

10043 OUTPUT FILES

10044 None.

10045 EXTENDED DESCRIPTION

10046 None.

10047 EXIT STATUS

10048 When the **-v** or **-V** options are specified, the following exit values shall be returned:

10049 0 Successful completion.

10050 >0 The *command_name* could not be found or an error occurred.

10051 Otherwise, the following exit values shall be returned:

10052 126 The utility specified by *command_name* was found but could not be invoked.

10053 127 An error occurred in the *command* utility or the utility specified by *command_name* could not be found.

10055 Otherwise, the exit status of *command* shall be that of the simple command specified by the arguments to *command*.

10057 CONSEQUENCES OF ERRORS

10058 Default.

10059 APPLICATION USAGE

10060 The order for command search allows functions to override regular built-ins and path searches.
10061 This utility is necessary to allow functions that have the same name as a utility to call the utility
10062 (instead of a recursive call to the function).

10063 The system default path is available using *getconf*, however, since *getconf* may need to have the
10064 *PATH* set up before it can be called itself, the following can be used:

10065 `command -p getconf _CS_PATH`

10066 There are some advantages to suppressing the special characteristics of special built-ins on
10067 occasion. For example:

10068 `command exec > unwritable-file`

10069 does not cause a non-interactive script to abort, so that the output status can be checked by the
10070 script.

10071 The *command*, *env*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if an
10072 error occurs so that applications can distinguish “failure to find a utility” from “invoked utility
10073 exited with an error indication”. The value 127 was chosen because it is not commonly used for
10074 other meanings; most utilities use small values for “normal error conditions” and the values
10075 above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen
10076 in a similar manner to indicate that the utility could be found, but not invoked. Some scripts
10077 produce meaningful error messages differentiating the 126 and 127 cases. The distinction
10078 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to
10079 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for
10080 any other reason.

10081 Since the **-v** and **-V** options of *command* produce output in relation to the current shell execution
10082 environment, *command* is generally provided as a shell regular built-in. If it is called in a subshell
10083 or separate utility execution environment, such as one of the following:

```
10084 (PATH=foo command -v)
10085 nohup command -v
```

10086 it does not necessarily produce correct results. For example, when called with *nohup* or an *exec*
10087 function, in a separate utility execution environment, most implementations are not able to
10088 identify aliases, functions, or special built-ins.

10089 Two types of regular built-ins could be encountered on a system and these are described
10090 separately by *command*. The description of command search in Section 2.9.1.1 (on page 48)
10091 allows for a standard utility to be implemented as a regular built-in as long as it is found in the
10092 appropriate place in a *PATH* search. So, for example, *command -v true* might yield */bin/true* or
10093 some similar pathname. Other implementation-defined utilities that are not defined by this
10094 volume of IEEE Std 1003.1-2001 might exist only as built-ins and have no pathname associated
10095 with them. These produce output identified as (regular) built-ins. Applications encountering
10096 these are not able to count on execing them, using them with *nohup*, overriding them with a
10097 different *PATH*, and so on.

10098 EXAMPLES

1. Make a version of *cd* that always prints out the new working directory exactly once:

```
10100 cd() {
10101     command cd "$@" >/dev/null
10102     pwd
10103 }
```

2. Start off a “secure shell script” in which the script avoids being spoofed by its parent:

```
10104 IFS='
10105 '
10106 #
10107 #      The preceding value should be <space><tab><newline>.
10108 #      Set IFS to its default value.

10109 \unalias -a
10110 #      Unset all possible aliases.
10111 #      Note that unalias is escaped to prevent an alias
10112 #      being used for unalias.

10113 unset -f command
10114 #      Ensure command is not a user function.

10115 PATH=$(command -p getconf _CS_PATH):$PATH"
10116 #      Put on a reliable PATH prefix.

10117 #      ...
```

10118 At this point, given correct permissions on the directories called by *PATH*, the script has
10119 the ability to ensure that any utility it calls is the intended one. It is being very cautious
10120 because it assumes that implementation extensions may be present that would allow user
10121 functions to exist when it is invoked; this capability is not specified by this volume of
10122 IEEE Std 1003.1-2001, but it is not prohibited as an extension. For example, the *ENV*
10123 variable precedes the invocation of the script with a user start-up script. Such a script
10124 could define functions to spoof the application.

10125 RATIONALE

10126 Since *command* is a regular built-in utility it is always found prior to the *PATH* search.

10127 There is nothing in the description of *command* that implies the command line is parsed any
10128 differently from that of any other simple command. For example:

10129 command a | b ; c
10130 is not parsed in any special way that causes ‘|’ or ‘;’ to be treated other than a pipe operator
10131 or semicolon or that prevents function lookup on **b** or **c**.

10132 The *command* utility is somewhat similar to the Eighth Edition shell *builtin* command, but since
10133 *command* also goes to the file system to search for utilities, the name *builtin* would not be
10134 intuitive.

10135 The *command* utility is most likely to be provided as a regular built-in. It is not listed as a special
10136 built-in for the following reasons:

- 10137 • The removal of exportable functions made the special precedence of a special built-in
10138 unnecessary.
- 10139 • A special built-in has special properties (see Section 2.14 (on page 64)) that were
10140 inappropriate for invoking other utilities. For example, two commands such as:

10141 date > *unwritable-file*
10142 command date > *unwritable-file*

10143 would have entirely different results; in a non-interactive script, the former would continue
10144 to execute the next command, the latter would abort. Introducing this semantic difference
10145 along with suppressing functions was seen to be non-intuitive.

10146 The –p option is present because it is useful to be able to ensure a safe path search that finds all
10147 the standard utilities. This search might not be identical to the one that occurs through one of the
10148 *exec* functions (as defined in the System Interfaces volume of IEEE Std 1003.1-2001) when *PATH*
10149 is unset. At the very least, this feature is required to allow the script to access the correct version
10150 of *getconf* so that the value of the default path can be accurately retrieved.

10151 The *command* –v and –V options were added to satisfy requirements from users that are
10152 currently accomplished by three different historical utilities: *type* in the System V shell, *whence* in
10153 the KornShell, and *which* in the C shell. Since there is no historical agreement on how and what
10154 to accomplish here, the POSIX *command* utility was enhanced and the historical utilities were left
10155 unmodified. The C shell *which* merely conducts a path search. The KornShell *whence* is more
10156 elaborate—in addition to the categories required by POSIX, it also reports on tracked aliases,
10157 exported aliases, and undefined functions.

10158 The output format of –V was left mostly unspecified because human users are its only audience.
10159 Applications should not be written to care about this information; they can use the output of –v
10160 to differentiate between various types of commands, but the additional information that may be
10161 emitted by the more verbose –V is not needed and should not be arbitrarily constrained in its
10162 verbosity or localization for application parsing reasons.

10163 FUTURE DIRECTIONS

10164 None.

10165 SEE ALSO

10166 Section 2.9.1.1 (on page 48), Section 2.12 (on page 61), Section 2.14 (on page 64), *sh*, *type*, the
10167 System Interfaces volume of IEEE Std 1003.1-2001, *exec*

10168 CHANGE HISTORY

10169 First released in Issue 4.

10170 NAME

10171 compress — compress data

10172 SYNOPSIS

10173 XSI compress [-fv][-b bits][file ...]

10174 compress [-cfv][-b bits][file]

10175

10176 DESCRIPTION

10177 The *compress* utility shall attempt to reduce the size of the named files by using adaptive
10178 Lempel-Ziv coding algorithm.

10179 Note: Lempel-Ziv is US Patent 4464650, issued to William Eastman, Abraham Lempel, Jacob Ziv,
10180 Martin Cohn on August 7th, 1984, and assigned to Sperry Corporation.

10181 Lempel-Ziv-Welch compression is covered by US Patent 4558302, issued to Terry A. Welch on
10182 December 10th, 1985, and assigned to Sperry Corporation.

10183 On systems not supporting adaptive Lempel-Ziv coding algorithm, the input files shall not be
10184 changed and an error value greater than two shall be returned. Except when the output is to the
10185 standard output, each file shall be replaced by one with the extension .Z. If the invoking process
10186 has appropriate privileges, the ownership, modes, access time, and modification time of the
10187 original file are preserved. If appending the .Z to the filename would make the name exceed
10188 {NAME_MAX} bytes, the command shall fail. If no files are specified, the standard input shall be
10189 compressed to the standard output.

10190 OPTIONS

10191 The *compress* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001,
10192 Section 12.2, Utility Syntax Guidelines.

10193 The following options shall be supported:

10194 **-b bits** Specify the maximum number of bits to use in a code. For a conforming
10195 application, the *bits* argument shall be:

10196 9 <= *bits* <= 14

10197 The implementation may allow *bits* values of greater than 14. The default is 14, 15,
10198 or 16.

10199 **-c** Cause *compress* to write to the standard output; the input file is not changed, and
10200 no .Z files are created.

10201 **-f** Force compression of *file*, even if it does not actually reduce the size of the file, or if
10202 the corresponding *file.Z* file already exists. If the **-f** option is not given, and the
10203 process is not running in the background, the user is prompted as to whether an
10204 existing *file.Z* file should be overwritten.

10205 **-v** Write the percentage reduction of each file to standard error.

10206 OPERANDS

10207 The following operand shall be supported:

10208 *file* A pathname of a file to be compressed.

10209 STDIN

10210 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-'.

10211 INPUT FILES

10212 If *file* operands are specified, the input files contain the data to be compressed.

10213 ENVIRONMENT VARIABLES

10214 The following environment variables shall affect the execution of *compress*:

10215 *LANG* Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

10219 *LC_ALL* If set to a non-empty string value, override the values of all the other
10220 internationalization variables.

10221 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
10222 characters (for example, single-byte as opposed to multi-byte characters in
10223 arguments).

10224 *LC_MESSAGES*

10225 Determine the locale that should be used to affect the format and contents of
10226 diagnostic messages written to standard error.

10227 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

10228 ASYNCHRONOUS EVENTS

10229 Default.

10230 STDOUT

10231 If no *file* operands are specified, or if a *file* operand is ‘–’, or if the **–c** option is specified, the
10232 standard output contains the compressed output.

10233 STDERR

10234 The standard error shall be used only for diagnostic and prompt messages and the output from
10235 **–v**.

10236 OUTPUT FILES

10237 The output files shall contain the compressed output. The format of compressed files is
10238 unspecified and interchange of such files between implementations (including access via
10239 unspecified file sharing mechanisms) is not required by IEEE Std 1003.1-2001.

10240 EXTENDED DESCRIPTION

10241 None.

10242 EXIT STATUS

10243 The following exit values shall be returned:

10244 0 Successful completion.

10245 1 An error occurred.

10246 2 One or more files were not compressed because they would have increased in size (and the
10247 **–f** option was not specified).

10248 >2 An error occurred.

10249 CONSEQUENCES OF ERRORS

10250 The input file shall remain unmodified.

10251 APPLICATION USAGE

10252 The amount of compression obtained depends on the size of the input, the number of *bits* per
10253 code, and the distribution of common substrings. Typically, text such as source code or English
10254 is reduced by 50-60%. Compression is generally much better than that achieved by Huffman
10255 coding or adaptive Huffman coding (*compact*), and takes less time to compute.

10256 Although *compress* strictly follows the default actions upon receipt of a signal or when an error
10257 occurs, some unexpected results may occur. In some implementations it is likely that a partially
10258 compressed file is left in place, alongside its uncompressed input file. Since the general
10259 operation of *compress* is to delete the uncompressed file only after the .Z file has been
10260 successfully filled, an application should always carefully check the exit status of *compress* before
10261 arbitrarily deleting files that have like-named neighbors with .Z suffixes.

10262 The limit of 14 on the *bits* option-argument is to achieve portability to all systems (within the
10263 restrictions imposed by the lack of an explicit published file format). Some implementations
10264 based on 16-bit architectures cannot support 15 or 16-bit uncompression.

10265 EXAMPLES

10266 None.

10267 RATIONALE

10268 None.

10269 FUTURE DIRECTIONS

10270 None.

10271 SEE ALSO

10272 *uncompress*, *zcat*

10273 CHANGE HISTORY

10274 First released in Issue 4.

10275 Issue 6

10276 The normative text is reworded to avoid use of the term “must” for application requirements.

10277 An error case is added for systems not supporting adaptive Lempel-Ziv coding.

10278 NAME

10279 cp — copy files

10280 SYNOPSIS

```
10281     cp [-fip] source_file target_file
10282     cp [-fip] source_file ... target
10283     cp -R [-H | -L | -P][-fip] source_file ... target
10284 OB    cp -r [-H | -L | -P][-fip] source_file ... target
```

10285 DESCRIPTION

10286 The first synopsis form is denoted by two operands, neither of which are existing files of type
 10287 directory. The *cp* utility shall copy the contents of *source_file* (or, if *source_file* is a file of type
 10288 symbolic link, the contents of the file referenced by *source_file*) to the destination path named by
 10289 *target_file*.

10290 The second synopsis form is denoted by two or more operands where the **-R** or **-r** options are
 10291 not specified and the first synopsis form is not applicable. It shall be an error if any *source_file* is a
 10292 file of type directory, if *target* does not exist, or if *target* is a file of a type defined by the System
 10293 Interfaces volume of IEEE Std 1003.1-2001, but is not a file of type directory. The *cp* utility shall
 10294 copy the contents of each *source_file* (or, if *source_file* is a file of type symbolic link, the contents
 10295 of the file referenced by *source_file*) to the destination path named by the concatenation of *target*,
 10296 a slash character, and the last component of *source_file*.

10297 The third and fourth synopsis forms are denoted by two or more operands where the **-R** or **-r**
 10298 options are specified. The *cp* utility shall copy each file in the file hierarchy rooted in each
 10299 *source_file* to a destination path named as follows:

- 10300 • If *target* exists and is a file of type directory, the name of the corresponding destination path
 10301 for each file in the file hierarchy shall be the concatenation of *target*, a slash character, and the
 10302 pathname of the file relative to the directory containing *source_file*.
- 10303 • If *target* does not exist and two operands are specified, the name of the corresponding destination path
 10304 for *source_file* shall be *target*; the name of the corresponding destination path
 10305 for all other files in the file hierarchy shall be the concatenation of *target*, a slash character,
 10306 and the pathname of the file relative to *source_file*.

10307 It shall be an error if *target* does not exist and more than two operands are specified, or if *target*
 10308 exists and is a file of a type defined by the System Interfaces volume of IEEE Std 1003.1-2001, but
 10309 is not a file of type directory.

10310 In the following description, the term *dest_file* refers to the file named by the destination path.
 10311 The term *source_file* refers to the file that is being copied, whether specified as an operand or a
 10312 file in a file hierarchy rooted in a *source_file* operand. If *source_file* is a file of type symbolic link:

- 10313 • If neither the **-R** nor **-r** options were specified, *cp* shall take actions based on the type and
 10314 contents of the file referenced by the symbolic link, and not by the symbolic link itself.
- 10315 • If the **-R** option was specified:
 - 10316 — If none of the options **-H**, **-L**, nor **-P** were specified, it is unspecified which of **-H**, **-L**, or
 10317 **-P** will be used as a default.
 - 10318 — If the **-H** option was specified, *cp* shall take actions based on the type and contents of the
 10319 file referenced by any symbolic link specified as a *source_file* operand.
 - 10320 — If the **-L** option was specified, *cp* shall take actions based on the type and contents of the
 10321 file referenced by any symbolic link specified as a *source_file* operand or any symbolic

- links encountered during traversal of a file hierarchy.
- If the **-P** option was specified, *cp* shall copy any symbolic link specified as a *source_file* operand and any symbolic links encountered during traversal of a file hierarchy, and shall not follow any symbolic links.
 - If the **-r** option was specified, the behavior is implementation-defined.
- For each *source_file*, the following steps shall be taken:
1. If *source_file* references the same file as *dest_file*, *cp* may write a diagnostic message to standard error; it shall do nothing more with *source_file* and shall go on to any remaining files.
 2. If *source_file* is of type directory, the following steps shall be taken:
 - a. If neither the **-R** or **-r** options were specified, *cp* shall write a diagnostic message to standard error, do nothing more with *source_file*, and go on to any remaining files.
 - b. If *source_file* was not specified as an operand and *source_file* is dot or dot-dot, *cp* shall do nothing more with *source_file* and go on to any remaining files.
 - c. If *dest_file* exists and it is a file type not specified by the System Interfaces volume of IEEE Std 1003.1-2001, the behavior is implementation-defined.
 - d. If *dest_file* exists and it is not of type directory, *cp* shall write a diagnostic message to standard error, do nothing more with *source_file* or any files below *source_file* in the file hierarchy, and go on to any remaining files.
 - e. If the directory *dest_file* does not exist, it shall be created with file permission bits set to the same value as those of *source_file*, modified by the file creation mask of the user if the **-p** option was not specified, and then bitwise-inclusively OR'ed with S_IRWXU. If *dest_file* cannot be created, *cp* shall write a diagnostic message to standard error, do nothing more with *source_file*, and go on to any remaining files. It is unspecified if *cp* attempts to copy files in the file hierarchy rooted in *source_file*.
 - f. The files in the directory *source_file* shall be copied to the directory *dest_file*, taking the four steps (1 to 4) listed here with the files as *source_files*.
 - g. If *dest_file* was created, its file permission bits shall be changed (if necessary) to be the same as those of *source_file*, modified by the file creation mask of the user if the **-p** option was not specified.
 - h. The *cp* utility shall do nothing more with *source_file* and go on to any remaining files.
 3. If *source_file* is of type regular file, the following steps shall be taken:
 - a. If *dest_file* exists, the following steps shall be taken:
 - i. If the **-i** option is in effect, the *cp* utility shall write a prompt to the standard error and read a line from the standard input. If the response is not affirmative, *cp* shall do nothing more with *source_file* and go on to any remaining files.
 - ii. A file descriptor for *dest_file* shall be obtained by performing actions equivalent to the *open()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001 called using *dest_file* as the *path* argument, and the bitwise-inclusive OR of O_WRONLY and O_TRUNC as the *oflag* argument.
 - iii. If the attempt to obtain a file descriptor fails and the **-f** option is in effect, *cp* shall attempt to remove the file by performing actions equivalent to the *unlink()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001 called using *dest_file* as the *path* argument.

- 10365 IEEE Std 1003.1-2001 called using *dest_file* as the *path* argument. If this attempt
10366 succeeds, *cp* shall continue with step 3b.
- 10367 b. If *dest_file* does not exist, a file descriptor shall be obtained by performing actions
10368 equivalent to the *open()* function defined in the System Interfaces volume of
10369 IEEE Std 1003.1-2001 called using *dest_file* as the *path* argument, and the bitwise-
10370 inclusive OR of O_WRONLY and O_CREAT as the *oflag* argument. The file
10371 permission bits of *source_file* shall be the *mode* argument.
- 10372 c. If the attempt to obtain a file descriptor fails, *cp* shall write a diagnostic message to
10373 standard error, do nothing more with *source_file*, and go on to any remaining files.
- 10374 d. The contents of *source_file* shall be written to the file descriptor. Any write errors
10375 shall cause *cp* to write a diagnostic message to standard error and continue to step 3e.
- 10376 e. The file descriptor shall be closed.
- 10377 f. The *cp* utility shall do nothing more with *source_file*. If a write error occurred in step
10378 3d, it is unspecified if *cp* continues with any remaining files. If no write error
10379 occurred in step 3d, *cp* shall go on to any remaining files.
- 10380 4. Otherwise, the following steps shall be taken:
- 10381 a. If the **-r** option was specified, the behavior is implementation-defined.
- 10382 b. If the **-R** option was specified, the following steps shall be taken:
- 10383 i. The *dest_file* shall be created with the same file type as *source_file*.
- 10384 ii. If *source_file* is a file of type FIFO, the file permission bits shall be the same as
10385 those of *source_file*, modified by the file creation mask of the user if the **-p**
10386 option was not specified. Otherwise, the permissions, owner ID, and group ID
10387 of *dest_file* are implementation-defined.
- 10388 If this creation fails for any reason, *cp* shall write a diagnostic message to
10389 standard error, do nothing more with *source_file*, and go on to any remaining files.
- 10390 iii. If *source_file* is a file of type symbolic link, the pathname contained in *dest_file*
10391 shall be the same as the pathname contained in *source_file*.
- 10392 If this fails for any reason, *cp* shall write a diagnostic message to standard error,
10393 do nothing more with *source_file*, and go on to any remaining files.
- 10394 If the implementation provides additional or alternate access control mechanisms (see the Base
10395 Definitions volume of IEEE Std 1003.1-2001, Section 4.4, File Access Permissions), their effect on
10396 copies of files is implementation-defined.

10398 OPTIONS

10399 The *cp* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
10400 Utility Syntax Guidelines.

10401 The following options shall be supported:

- 10402 **-f** If a file descriptor for a destination file cannot be obtained, as described in step
10403 3.a.ii., attempt to unlink the destination file and proceed.
- 10404 **-H** Take actions based on the type and contents of the file referenced by any symbolic
10405 link specified as a *source_file* operand.
- 10406 **-i** Write a prompt to standard error before copying to any existing destination file. If
10407 the response from the standard input is affirmative, the copy shall be attempted;

10408 otherwise, it shall not.

10409 **-L** Take actions based on the type and contents of the file referenced by any symbolic link specified as a *source_file* operand or any symbolic links encountered during traversal of a file hierarchy.

10410 **-P** Take actions on any symbolic link specified as a *source_file* operand or any symbolic link encountered during traversal of a file hierarchy.

10411 **-p** Duplicate the following characteristics of each source file in the corresponding destination file:

1. The time of last data modification and time of last access. If this duplication fails for any reason, *cp* shall write a diagnostic message to standard error.
2. The user ID and group ID. If this duplication fails for any reason, it is unspecified whether *cp* writes a diagnostic message to standard error.
3. The file permission bits and the S_ISUID and S_ISGID bits. Other, implementation-defined, bits may be duplicated as well. If this duplication fails for any reason, *cp* shall write a diagnostic message to standard error.

10412 If the user ID or the group ID cannot be duplicated, the file permission bits S_ISUID and S_ISGID shall be cleared. If these bits are present in the source file but are not duplicated in the destination file, it is unspecified whether *cp* writes a diagnostic message to standard error.

10413 The order in which the preceding characteristics are duplicated is unspecified. The *dest_file* shall not be deleted if these characteristics cannot be preserved.

10414 **-R** Copy file hierarchies.

10415 OB **-r** Copy file hierarchies. The treatment of special files is implementation-defined.

10416 Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be considered an error. The last option specified shall determine the behavior of the utility.

10433 OPERANDS

10434 The following operands shall be supported:

10435 *source_file* A pathname of a file to be copied.

10436 *target_file* A pathname of an existing or nonexistent file, used for the output when a single file is copied.

10437 *target* A pathname of a directory to contain the copied files.

10439 STDIN

10440 The standard input shall be used to read an input line in response to each prompt specified in the STDERR section. Otherwise, the standard input shall not be used.

10442 INPUT FILES

10443 The input files specified as operands may be of any file type.

10444 ENVIRONMENT VARIABLES

10445 The following environment variables shall affect the execution of *cp*:

10446 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

10450	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
10452	<i>LC_COLLATE</i>	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the extended regular expression defined for the yesexpr locale keyword in the <i>LC_MESSAGES</i> category.
10456	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes used in the extended regular expression defined for the yesexpr locale keyword in the <i>LC_MESSAGES</i> category.
10461	<i>LC_MESSAGES</i>	Determine the locale for the processing of affirmative responses that should be used to affect the format and contents of diagnostic messages written to standard error.
10465 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
10466	ASYNCHRONOUS EVENTS	
10467		Default.
10468	STDOUT	
10469		Not used.
10470	STDERR	
10471		A prompt shall be written to standard error under the conditions specified in the DESCRIPTION section. The prompt shall contain the destination pathname, but its format is otherwise unspecified. Otherwise, the standard error shall be used only for diagnostic messages.
10474	OUTPUT FILES	
10475		The output files may be of any type.
10476	EXTENDED DESCRIPTION	
10477		None.
10478	EXIT STATUS	
10479		The following exit values shall be returned:
10480	0	All files were copied successfully.
10481	>0	An error occurred.
10482	CONSEQUENCES OF ERRORS	
10483		If <i>cp</i> is prematurely terminated by a signal or error, files or file hierarchies may be only partially copied and files and directories may have incorrect permissions or access and modification times.

10486 APPLICATION USAGE

10487 The difference between **-R** and **-r** is in the treatment by *cp* of file types other than regular and
10488 directory. The original **-r** flag, for historic reasons, does not handle special files any differently
10489 from regular files, but always reads the file and copies its contents. This has obvious problems in
10490 the presence of special file types; for example, character devices, FIFOs, and sockets. The **-R**
10491 option is intended to recreate the file hierarchy and the **-r** option supports historical practice. It
10492 was anticipated that a future version of this volume of IEEE Std 1003.1-2001 would deprecate the
10493 **-r** option, and for that reason, there has been no attempt to fix its behavior with respect to FIFOs
10494 or other file types where copying the file is clearly wrong. However, some implementations
10495 support **-r** with the same abilities as the **-R** defined in this volume of IEEE Std 1003.1-2001. To
10496 accommodate them as well as systems that do not, the differences between **-r** and **-R** are
10497 implementation-defined. Implementations may make them identical. The **-r** option is marked
10498 obsolescent.

10499 The set-user-ID and set-group-ID bits are explicitly cleared when files are created. This is to
10500 prevent users from creating programs that are set-user-ID or set-group-ID to them when
10501 copying files or to make set-user-ID or set-group-ID files accessible to new groups of users. For
10502 example, if a file is set-user-ID and the copy has a different group ID than the source, a new
10503 group of users has execute permission to a set-user-ID program than did previously. In
10504 particular, this is a problem for superusers copying users' trees.

10505 EXAMPLES

10506 None.

10507 RATIONALE

10508 The **-i** option exists on BSD systems, giving applications and users a way to avoid accidentally
10509 removing files when copying. Although the 4.3 BSD version does not prompt if the standard
10510 input is not a terminal, the standard developers decided that use of **-i** is a request for interaction,
10511 so when the destination path exists, the utility takes instructions from whatever responds on
10512 standard input.

10513 The exact format of the interactive prompts is unspecified. Only the general nature of the
10514 contents of prompts are specified because implementations may desire more descriptive
10515 prompts than those used on historical implementations. Therefore, an application using the **-i**
10516 option relies on the system to provide the most suitable dialog directly with the user, based on
10517 the behavior specified.

10518 The **-p** option is historical practice on BSD systems, duplicating the time of last data
10519 modification and time of last access. This volume of IEEE Std 1003.1-2001 extends it to preserve
10520 the user and group IDs, as well as the file permissions. This requirement has obvious problems
10521 in that the directories are almost certainly modified after being copied. This volume of
10522 IEEE Std 1003.1-2001 requires that the modification times be preserved. The statement that the
10523 order in which the characteristics are duplicated is unspecified is to permit implementations to
10524 provide the maximum amount of security for the user. Implementations should take into
10525 account the obvious security issues involved in setting the owner, group, and mode in the
10526 wrong order or creating files with an owner, group, or mode different from the final value.

10527 It is unspecified whether *cp* writes diagnostic messages when the user and group IDs cannot be
10528 set due to the widespread practice of users using **-p** to duplicate some portion of the file
10529 characteristics, indifferent to the duplication of others. Historic implementations only write
10530 diagnostic messages on errors other than [EPERM].

10531 The **-r** option is historical practice on BSD and BSD-derived systems, copying file hierarchies as
10532 opposed to single files. This functionality is used heavily in historical applications, and its loss
10533 would significantly decrease consensus. The **-R** option was added as a close synonym to the **-r**
10534 option, selected for consistency with all other options in this volume of IEEE Std 1003.1-2001 that

10535 do recursive directory descent.

10536 When a failure occurs during the copying of a file hierarchy, *cp* is required to attempt to copy
10537 files that are on the same level in the hierarchy or above the file where the failure occurred. It is
10538 unspecified if *cp* shall attempt to copy files below the file where the failure occurred (which
10539 cannot succeed in any case).

10540 Permissions, owners, and groups of created special file types have been deliberately left as
10541 implementation-defined. This is to allow systems to satisfy special requirements (for example,
10542 allowing users to create character special devices, but requiring them to be owned by a certain
10543 group). In general, it is strongly suggested that the permissions, owner, and group be the same
10544 as if the user had run the historical *mknod*, *In*, or other utility to create the file. It is also probable
10545 that additional privileges are required to create block, character, or other implementation-
10546 defined special file types.

10547 Additionally, the **-p** option explicitly requires that all set-user-ID and set-group-ID permissions
10548 be discarded if any of the owner or group IDs cannot be set. This is to keep users from
10549 unintentionally giving away special privilege when copying programs.

10550 When creating regular files, historical versions of *cp* use the mode of the source file as modified
10551 by the file mode creation mask. Other choices would have been to use the mode of the source file
10552 unmodified by the creation mask or to use the same mode as would be given to a new file
10553 created by the user (plus the execution bits of the source file) and then modify it by the file mode
10554 creation mask. In the absence of any strong reason to change historic practice, it was in large part
10555 retained.

10556 When creating directories, historical versions of *cp* use the mode of the source directory, plus
10557 read, write, and search bits for the owner, as modified by the file mode creation mask. This is
10558 done so that *cp* can copy trees where the user has read permission, but the owner does not. A
10559 side effect is that if the file creation mask denies the owner permissions, *cp* fails. Also, once the
10560 copy is done, historical versions of *cp* set the permissions on the created directory to be the same
10561 as the source directory, unmodified by the file creation mask.

10562 This behavior has been modified so that *cp* is always able to create the contents of the directory,
10563 regardless of the file creation mask. After the copy is done, the permissions are set to be the same
10564 as the source directory, as modified by the file creation mask. This latter change from historical
10565 behavior is to prevent users from accidentally creating directories with permissions beyond
10566 those they would normally set and for consistency with the behavior of *cp* in creating files.

10567 It is not a requirement that *cp* detect attempts to copy a file to itself; however, implementations
10568 are strongly encouraged to do so. Historical implementations have detected the attempt in most
10569 cases.

10570 There are two methods of copying subtrees in this volume of IEEE Std 1003.1-2001. The other
10571 method is described as part of the *pax* utility (see *pax*). Both methods are historical practice. The
10572 *cp* utility provides a simpler, more intuitive interface, while *pax* offers a finer granularity of
10573 control. Each provides additional functionality to the other; in particular, *pax* maintains the
10574 hard-link structure of the hierarchy, while *cp* does not. It is the intention of the standard
10575 developers that the results be similar (using appropriate option combinations in both utilities).
10576 The results are not required to be identical; there seemed insufficient gain to applications to
10577 balance the difficulty of implementations having to guarantee that the results would be exactly
10578 identical.

10579 The wording allowing *cp* to copy a directory to implementation-defined file types not specified
10580 by the System Interfaces volume of IEEE Std 1003.1-2001 is provided so that implementations
10581 supporting symbolic links are not required to prohibit copying directories to symbolic links.
10582 Other extensions to the System Interfaces volume of IEEE Std 1003.1-2001 file types may need to

10583 use this loophole as well.

10584 FUTURE DIRECTIONS

10585 The **-r** option may be removed; use **-R** instead.

10586 SEE ALSO

10587 *mv, find, ln, pax, the *(Zy, open(), unlink()*

10588 CHANGE HISTORY

10589 First released in Issue 2.

10590 Issue 6

10591 The **-r** option is marked obsolescent.

10592 The new options **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These
10593 options affect the processing of symbolic links.

10594 IEEE PASC Interpretation 1003.2 #194 is applied, adding a description of the **-P** option.

10595 NAME

10596 crontab — schedule periodic background work

10597 SYNOPSIS

10598 UP **crontab [file]**

10599 **crontab [-e | -l | -r]**

10600

10601 DESCRIPTION

10602 The *crontab* utility shall create, replace, or edit a user's crontab entry; a crontab entry is a list of
10603 commands and the times at which they shall be executed. The new crontab entry can be input by
10604 specifying *file* or input from standard input if no *file* operand is specified, or by using an editor, if
10605 **-e** is specified.

10606 Upon execution of a command from a crontab entry, the implementation shall supply a default
10607 environment, defining at least the following environment variables:

10608 **HOME** A pathname of the user's home directory.

10609 **LOGNAME** The user's login name.

10610 **PATH** A string representing a search path guaranteed to find all of the standard utilities.

10611 **SHELL** A pathname of the command interpreter. When *crontab* is invoked as specified by
10612 this volume of IEEE Std 1003.1-2001, the value shall be a pathname for *sh*.

10613 The values of these variables when *crontab* is invoked as specified by this volume of
10614 IEEE Std 1003.1-2001 shall not affect the default values provided when the scheduled command
10615 is run.

10616 If standard output and standard error are not redirected by commands executed from the
10617 crontab entry, any generated output or errors shall be mailed, via an implementation-defined
10618 method, to the user.

10619 XSI Users shall be permitted to use *crontab* if their names appear in the file **/usr/lib/cron/cron.allow**.
10620 If that file does not exist, the file **/usr/lib/cron/cron.deny** shall be checked to determine whether
10621 the user shall be denied access to *crontab*. If neither file exists, only a process with appropriate
10622 privileges shall be allowed to submit a job. If only **cron.deny** exists and is empty, global usage
10623 shall be permitted. The **cron.allow** and **cron.deny** files shall consist of one user name per line.

10624 OPTIONS

10625 The *crontab* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
10626 12.2, Utility Syntax Guidelines.

10627 The following options shall be supported:

10628 **-e** Edit a copy of the invoking user's crontab entry, or create an empty entry to edit if
10629 the crontab entry does not exist. When editing is complete, the entry shall be
10630 installed as the user's crontab entry.

10631 **-l** (The letter ell.) List the invoking user's crontab entry.

10632 **-r** Remove the invoking user's crontab entry.

10633 OPERANDS

10634 The following operand shall be supported:

10635 **file** The pathname of a file that contains specifications, in the format defined in the
10636 INPUT FILES section, for crontab entries.

10637 **STDIN**

10638 See the INPUT FILES section.

10639 **INPUT FILES**10640 In the POSIX locale, the user or application shall ensure that a crontab entry is a text file
10641 consisting of lines of six fields each. The fields shall be separated by <blank>s. The first five
10642 fields shall be integer patterns that specify the following:

- 10643 1. Minute [0,59]
- 10644 2. Hour [0,23]
- 10645 3. Day of the month [1,31]
- 10646 4. Month of the year [1,12]
- 10647 5. Day of the week ([0,6] with 0=Sunday)

10648 Each of these patterns can be either an asterisk (meaning all valid values), an element, or a list of
10649 elements separated by commas. An element shall be either a number or two numbers separated
10650 by a hyphen (meaning an inclusive range). The specification of days can be made by two fields
10651 (day of the month and day of the week). If month, day of month, and day of week are all
10652 asterisks, every day shall be matched. If either the month or day of month is specified as an
10653 element or list, but the day of week is an asterisk, the month and day of month fields shall
10654 specify the days that match. If both month and day of month are specified as an asterisk, but day
10655 of week is an element or list, then only the specified days of the week match. Finally, if either the
10656 month or day of month is specified as an element or list, and the day of week is also specified as
10657 an element or list, then any day matching either the month and day of month, or the day of
10658 week, shall be matched.

10659 The sixth field of a line in a crontab entry is a string that shall be executed by *sh* at the specified
10660 times. A percent sign character in this field shall be translated to a <newline>. Any character
10661 preceded by a backslash (including the '%'') shall cause that character to be treated literally.
10662 Only the first line (up to a '%' or end-of-line) of the command field shall be executed by the
10663 command interpreter. The other lines shall be made available to the command as standard input.

10664 Blank lines and those whose first non-<blank> is '#' shall be ignored.

10665 XSI The text files */usr/lib/cron/cron.allow* and */usr/lib/cron/cron.deny* shall contain zero or more
10666 user names, one per line, of users who are, respectively, authorized or denied access to the
10667 service underlying the *crontab* utility.

10668 **ENVIRONMENT VARIABLES**10669 The following environment variables shall affect the execution of *crontab*:

- 10670 **EDITOR** Determine the editor to be invoked when the **-e** option is specified. The default
10671 editor shall be *vi*.
- 10672 **LANG** Provide a default value for the internationalization variables that are unset or null.
10673 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
10674 Internationalization Variables for the precedence of internationalization variables
10675 used to determine the values of locale categories.)
- 10676 **LC_ALL** If set to a non-empty string value, override the values of all the other
10677 internationalization variables.
- 10678 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
10679 characters (for example, single-byte as opposed to multi-byte characters in
10680 arguments and input files).

- 10681 ***LC_MESSAGES***
10682 Determine the locale that should be used to affect the format and contents of
10683 diagnostic messages written to standard error.
- 10684 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of ***LC_MESSAGES***.
- 10685 **ASYNCHRONOUS EVENTS**
- 10686 Default.
- 10687 **STDOUT**
- 10688 If the **-l** option is specified, the crontab entry shall be written to the standard output.
- 10689 **STDERR**
- 10690 The standard error shall be used only for diagnostic messages.
- 10691 **OUTPUT FILES**
- 10692 None.
- 10693 **EXTENDED DESCRIPTION**
- 10694 None.
- 10695 **EXIT STATUS**
- 10696 The following exit values shall be returned:
- 10697 0 Successful completion.
- 10698 >0 An error occurred.
- 10699 **CONSEQUENCES OF ERRORS**
- 10700 The user's crontab entry is not submitted, removed, edited, or listed.
- 10701 **APPLICATION USAGE**
- 10702 The format of the crontab entry shown here is guaranteed only for the POSIX locale. Other
10703 cultures may be supported with substantially different interfaces, although implementations are
10704 encouraged to provide comparable levels of functionality.
- 10705 The default settings of the *HOME*, *LOGNAME*, *PATH*, and *SHELL* variables that are given to the
10706 scheduled job are not affected by the settings of those variables when *crontab* is run; as stated,
10707 they are defaults. The text about "invoked as specified by this volume of IEEE Std 1003.1-2001"
10708 means that the implementation may provide extensions that allow these variables to be affected
10709 at runtime, but that the user has to take explicit action in order to access the extension, such as
10710 give a new option flag or modify the format of the crontab entry.
- 10711 A typical user error is to type only *crontab*; this causes the system to wait for the new crontab
10712 entry on standard input. If end-of-file is typed (generally <control>-D), the crontab entry is
10713 replaced by an empty file. In this case, the user should type the interrupt character, which
10714 prevents the crontab entry from being replaced.
- 10715 **EXAMPLES**
- 10716 1. Clean up **core** files every weekday morning at 3:15 am:
10717 15 3 * * 1-5 find \$HOME -name core 2>/dev/null | xargs rm -f
- 10718 2. Mail a birthday greeting:
10719 0 12 14 2 * mailx john%Happy Birthday!%Time for lunch.
- 10720 3. As an example of specifying the two types of days:
10721 0 0 1,15 * 1

10722 would run a command on the first and fifteenth of each month, as well as on every
10723 Monday. To specify days by only one field, the other field should be set to '*' ; for
10724 example:

10725 0 0 * * 1

10726 would run a command only on Mondays.

10727 RATIONALE

10728 All references to a *cron* daemon and to *cron files* have been omitted. Although historical
10729 implementations have used this arrangement, there is no reason to limit future implementations.

10730 This description of *crontab* is designed to support only users with normal privileges. The format
10731 of the input is based on the System V *crontab*; however, there is no requirement here that the
10732 actual system database used by the *cron* daemon (or a similar mechanism) use this format
10733 internally. For example, systems derived from BSD are likely to have an additional field
10734 appended that indicates the user identity to be used when the job is submitted.

10735 The **-e** option was adopted from the SVID as a user convenience, although it does not exist in all
10736 historical implementations.

10737 FUTURE DIRECTIONS

10738 None.

10739 SEE ALSO

10740 *at*

10741 CHANGE HISTORY

10742 First released in Issue 2.

10743 Issue 6

10744 This utility is marked as part of the User Portability Utilities option.

10745 The normative text is reworded to avoid use of the term "must" for application requirements.

10746 NAME

10747 csplit — split files based on context

10748 SYNOPSIS

10749 UP csplit [-ks][-f prefix][-n number] file arg1 ...argn

10750

10751 DESCRIPTION

10752 The *csplit* utility shall read the file named by the *file* operand, write all or part of that file into
10753 other files as directed by the *arg* operands, and write the sizes of the files.

10754 OPTIONS

10755 The *csplit* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
10756 12.2, Utility Syntax Guidelines.

10757 The following options shall be supported:

- 10758 **-f prefix** Name the created files *prefix00*, *prefix01*, ..., *prefixn*. The default is *xx00* ... *xxn*. If
10759 the *prefix* argument would create a filename exceeding {NAME_MAX} bytes, an
10760 error shall result, *csplit* shall exit with a diagnostic message, and no files shall be
10761 created.
- 10762 **-k** Leave previously created files intact. By default, *csplit* shall remove created files if
10763 an error occurs.
- 10764 **-n number** Use *number* decimal digits to form filenames for the file pieces. The default shall be
10765 2.
- 10766 **-s** Suppress the output of file size messages.

10767 OPERANDS

10768 The following operands shall be supported:

- 10769 **file** The pathname of a text file to be split. If *file* is '−', the standard input shall be
10770 used.

10771 The operands *arg1* ... *argn* can be a combination of the following:

- 10772 **/rexp/[offset]** A file shall be created using the content of the lines from the current line up to, but
10773 not including, the line that results from the evaluation of the regular expression
10774 with *offset*, if any, applied. The regular expression *rexp* shall follow the rules for
10775 basic regular expressions described in the Base Definitions volume of
10776 IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions. The application shall
10777 use the sequence "\/" to specify a slash character within the *rexp*. The optional
10778 offset shall be a positive or negative integer value representing a number of lines.
10779 A positive integer value can be preceded by '+'. If the selection of lines from an
10780 *offset* expression of this type would create a file with zero lines, or one with greater
10781 than the number of lines left in the input file, the results are unspecified. After the
10782 section is created, the current line shall be set to the line that results from the
10783 evaluation of the regular expression with any offset applied. If the current line is
10784 the first line in the file and a regular expression operation has not yet been
10785 performed, the pattern match of *rexp* shall be applied from the current line to the
10786 end of the file. Otherwise, the pattern match of *rexp* shall be applied from the line
10787 following the current line to the end of the file.

- 10788 **%rexp%[offset]** Equivalent to **/rexp/[offset]**, except that no file shall be created for the selected
10789 section of the input file. The application shall use the sequence "\%" to specify a

10792		percent-sign character within the <i>rexp</i> .
10793	<i>line_no</i>	Create a file from the current line up to (but not including) the line number <i>line_no</i> . Lines in the file shall be numbered starting at one. The current line becomes <i>line_no</i> .
10794		
10795		
10796	{ <i>num</i> }	Repeat operand. This operand can follow any of the operands described previously. If it follows a <i>rexp</i> type operand, that operand shall be applied <i>num</i> more times. If it follows a <i>line_no</i> operand, the file shall be split every <i>line_no</i> lines, <i>num</i> times, from that point.
10797		
10798		
10799		
10800		An error shall be reported if an operand does not reference a line between the current position and the end of the file.
10801		
10802	STDIN	
10803		See the INPUT FILES section.
10804	INPUT FILES	
10805		The input file shall be a text file.
10806	ENVIRONMENT VARIABLES	
10807		The following environment variables shall affect the execution of <i>csplit</i> :
10808	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
10809		
10810		
10811		
10812	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
10813		
10814	<i>LC_COLLATE</i>	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.
10815		
10816		
10817	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes within regular expressions.
10818		
10819		
10820		
10821	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
10822		
10823		
10824	XSI <i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
10825	ASYNCHRONOUS EVENTS	
10826		If the -k option is specified, created files shall be retained. Otherwise, the default action occurs.
10827	STDOUT	
10828		Unless the -s option is used, the standard output shall consist of one line per file created, with a format as follows:
10829		
10830		"%d\n", <file size in bytes>
10831	STDERR	
10832		The standard error shall be used only for diagnostic messages.

10833 OUTPUT FILES

10834 The output files shall contain portions of the original input file; otherwise, unchanged.

10835 EXTENDED DESCRIPTION

10836 None.

10837 EXIT STATUS

10838 The following exit values shall be returned:

10839 0 Successful completion.

10840 >0 An error occurred.

10841 CONSEQUENCES OF ERRORS

10842 By default, created files shall be removed if an error occurs. When the **-k** option is specified,
10843 created files shall not be removed if an error occurs.

10844 APPLICATION USAGE

10845 None.

10846 EXAMPLES

10847 1. This example creates four files, **cobol00** ... **cobol03**:

10848 `csplit -f cobol file '/procedure division/' /par5./ /par16./`

10849 After editing the split files, they can be recombined as follows:

10850 `cat cobol0[0-3] > file`

10851 Note that this example overwrites the original file.

10852 2. This example would split the file after the first 99 lines, and every 100 lines thereafter, up
10853 to 9 999 lines; this is because lines in the file are numbered from 1 rather than zero, for
10854 historical reasons:

10855 `csplit -k file 100 {99}`

10856 3. Assuming that **prog.c** follows the C-language coding convention of ending routines with a
10857 '}' at the beginning of the line, this example creates a file containing each separate C
10858 routine (up to 21) in **prog.c**:

10859 `csplit -k prog.c '%main(%' '/^}/+1' {20}`

10860 RATIONALE

10861 The **-n** option was added to extend the range of filenames that could be handled.

10862 Consideration was given to adding a **-a** flag to use the alphabetic filename generation used by
10863 the historical *split* utility, but the functionality added by the **-n** option was deemed to make
10864 alphabetic naming unnecessary.

10865 FUTURE DIRECTIONS

10866 None.

10867 SEE ALSO

10868 *sed*, *split*

10869 CHANGE HISTORY

10870 First released in Issue 2.

10871 Issue 5

10872 The FUTURE DIRECTIONS section is added.

10873 Issue 6

10874 This utility is marked as part of the User Portability Utilities option.

10875 The APPLICATION USAGE section is added.

10876 The description of regular expression operands is changed to align with the IEEE P1003.2b draft standard.
10877

10878 The normative text is reworded to avoid use of the term “must” for application requirements.

10879 **NAME**

10880 *ctags* — create a tags file (**DEVELOPMENT, FORTRAN**)

10881 **SYNOPSIS**

10882 UP *ctags [-a][-f tagsfile] pathname ...*

10883 *ctags -x pathname ...*

10884

10885 **DESCRIPTION**

10886 The *ctags* utility shall be provided on systems that support the User Portability Utilities option,
10887 the Software Development Utilities option, and either or both of the C-Language Development
10888 Utilities option and FORTRAN Development Utilities option. On other systems, it is optional.

10889 The *ctags* utility shall write a *tagsfile* or an index of objects from C-language or FORTRAN source
10890 files specified by the *pathname* operands. The *tagsfile* shall list the locators of language-specific
10891 objects within the source files. A locator consists of a name, pathname, and either a search
10892 pattern or a line number that can be used in searching for the object definition. The objects that
10893 shall be recognized are specified in the EXTENDED DESCRIPTION section.

10894 **OPTIONS**

10895 The *ctags* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
10896 12.2, Utility Syntax Guidelines.

10897 The following options shall be supported:

10898 **-a** Append to *tagsfile*.

10899 **-f tagsfile** Write the object locator lists into *tagsfile* instead of the default file named **tags** in
10900 the current directory.

10901 **-x** Produce a list of object names, the line number, and filename in which each is
10902 defined, as well as the text of that line, and write this to the standard output. A
10903 *tagsfile* shall not be created when **-x** is specified.

10904 **OPERANDS**

10905 The following *pathname* operands are supported:

10906 *file.c* Files with basenames ending with the **.c** suffix shall be treated as C-language
10907 source code. Such files that are not valid input to *c99* produce unspecified results.

10908 *file.h* Files with basenames ending with the **.h** suffix shall be treated as C-language
10909 source code. Such files that are not valid input to *c99* produce unspecified results.

10910 *file.f* Files with basenames ending with the **.f** suffix shall be treated as FORTRAN-
10911 language source code. Such files that are not valid input to *fort77* produce
10912 unspecified results.

10913 The handling of other files is implementation-defined.

10914 **STDIN**

10915 See the INPUT FILES section.

10916 **INPUT FILES**

10917 The input files shall be text files containing source code in the language indicated by the operand
10918 filename suffixes.

10919 ENVIRONMENT VARIABLES

10920 The following environment variables shall affect the execution of *ctags*:

10921 **LANG** Provide a default value for the internationalization variables that are unset or null.
10922 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
10923 Internationalization Variables for the precedence of internationalization variables
10924 used to determine the values of locale categories.)

10925 **LC_ALL** If set to a non-empty string value, override the values of all the other
10926 internationalization variables.

10927 **LC_COLLATE**

10928 Determine the order in which output is sorted for the **-x** option. The POSIX locale
10929 determines the order in which the *tagsfile* is written.

10930 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
10931 characters (for example, single-byte as opposed to multi-byte characters in
10932 arguments and input files). When processing C-language source code, if the locale
10933 is not compatible with the C locale described by the ISO C standard, the results are
10934 unspecified.

10935 **LC_MESSAGES**

10936 Determine the locale that should be used to affect the format and contents of
10937 diagnostic messages written to standard error.

10938 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

10939 ASYNCHRONOUS EVENTS

10940 Default.

10941 **STDOUT**

10942 The list of object name information produced by the **-x** option shall be written to standard
10943 output in the following format:

10944 "%s %d %s %s", <object-name>, <line-number>, <filename>, <text>

10945 where <text> is the text of line <line-number> of file <filename>.

10946 **STDERR**

10947 The standard error shall be used only for diagnostic messages.

10948 **OUTPUT FILES**

10949 When the **-x** option is not specified, the format of the output file shall be:

10950 "%s\t%s\t/%s/\n", <identifier>, <filename>, <pattern>

10951 where <pattern> is a search pattern that could be used by an editor to find the defining instance
10952 of <identifier> in <filename> (where *defining instance* is indicated by the declarations listed in the
10953 EXTENDED DESCRIPTION).

10954 An optional circumflex ('^') can be added as a prefix to <pattern>, and an optional dollar sign
10955 can be appended to <pattern> to indicate that the pattern is anchored to the beginning (end) of a
10956 line of text. Any slash or backslash characters in <pattern> shall be preceded by a backslash
10957 character. The anchoring circumflex, dollar sign, and escaping backslash characters shall not be
10958 considered part of the search pattern. All other characters in the search pattern shall be
10959 considered literal characters.

10960 An alternative format is:

10961 "%s\t%s\t?%s?\n", <identifier>, <filename>, <pattern>

10962 which is identical to the first format except that slashes in <pattern> shall not be preceded by
10963 escaping backslash characters, and question mark characters in <pattern> shall be preceded by
10964 backslash characters.

10965 A second alternative format is:

10966 "%s\t%s\t%d\n", <identifier>, <filename>, <lineno>

10967 where <lineno> is a decimal line number that could be used by an editor to find <identifier> in
10968 <filename>.

10969 Neither alternative format shall be produced by *ctags* when it is used as described by
10970 IEEE Std 1003.1-2001, but the standard utilities that process tags files shall be able to process
10971 those formats as well as the first format.

10972 In any of these formats, the file shall be sorted by identifier, based on the collation sequence in
10973 the POSIX locale.

10974 EXTENDED DESCRIPTION

10975 If the operand identifies C-language source, the *ctags* utility shall attempt to produce an output
10976 line for each of the following objects:

- 10977 • Function definitions
- 10978 • Type definitions
- 10979 • Macros with arguments

10980 It may also produce output for any of the following objects:

- 10981 • Function prototypes
- 10982 • Structures
- 10983 • Unions
- 10984 • Global variable definitions
- 10985 • Enumeration types
- 10986 • Macros without arguments
- 10987 • #define statements
- 10988 • #line statements

10989 Any #if and #ifdef statements shall produce no output. The tag **main** is treated specially in C
10990 programs. The tag formed shall be created by prefixing **M** to the name of the file, with the
10991 trailing .c, and leading pathname components (if any) removed.

10992 On systems that do not support the C-Language Development Utilities option, *ctags* produces
10993 unspecified results for C-language source code files. It should write to standard error a message
10994 identifying this condition and cause a non-zero exit status to be produced.

10995 If the operand identifies FORTRAN source, the *ctags* utility shall produce an output line for each
10996 function definition. It may also produce output for any of the following objects:

- 10997 • Subroutine definitions
- 10998 • COMMON statements

- 10999 • PARAMETER statements
 - 11000 • DATA and BLOCK DATA statements
 - 11001 • Statement numbers
- 11002 On systems that do not support the FORTRAN Development Utilities option, *ctags* produces
11003 unspecified results for FORTRAN source code files. It should write to standard error a message
11004 identifying this condition and cause a non-zero exit status to be produced.
- 11005 It is implementation-defined what other objects (including duplicate identifiers) produce output.
- 11006 **EXIT STATUS**
- 11007 The following exit values shall be returned:
- 11008 0 Successful completion.
 - 11009 >0 An error occurred.
- 11010 **CONSEQUENCES OF ERRORS**
- 11011 Default.
- 11012 **APPLICATION USAGE**
- 11013 The output with **-x** is meant to be a simple index that can be written out as an off-line readable
11014 function index. If the input files to *ctags* (such as .c files) were not created using the same locale
11015 as that in effect when *ctags -x* is run, results might not be as expected.
- 11016 The description of C-language processing says “attempts to” because the C language can be
11017 greatly confused, especially through the use of #defines, and this utility would be of no use if
11018 the real C preprocessor were run to identify them. The output from *ctags* may be fooled and
11019 incorrect for various constructs.
- 11020 **EXAMPLES**
- 11021 None.
- 11022 **RATIONALE**
- 11023 The option list was significantly reduced from that provided by historical implementations. The
11024 **-F** option was omitted as redundant, since it is the default. The **-B** option was omitted as being
11025 of very limited usefulness. The **-t** option was omitted since the recognition of **typedefs** is now
11026 required for C source files. The **-u** option was omitted because the update function was judged
11027 to be not only inefficient, but also rarely needed.
- 11028 An early proposal included a **-w** option to suppress warning diagnostics. Since the types of such
11029 diagnostics could not be described, the option was omitted as being not useful.
- 11030 The text for *LC_CTYPE* about compatibility with the C locale acknowledges that the ISO C
11031 standard imposes requirements on the locale used to process C source. This could easily be a
11032 superset of that known as “the C locale” by way of implementation extensions, or one of a few
11033 alternative locales for systems supporting different codesets. No statement is made for
11034 FORTRAN because the ANSI X3.9-1978 standard (FORTRAN 77) does not (yet) define a similar
11035 locale concept. However, a general rule in this volume of IEEE Std 1003.1-2001 is that any time
11036 that locales do not match (preparing a file for one locale and processing it in another), the results
11037 are suspect.
- 11038 The collation sequence of the tags file is not affected by *LC_COLLATE* because it is typically not
11039 used by human readers, but only by programs such as *vi* to locate the tag within the source files.
11040 Using the POSIX locale eliminates some of the problems of coordinating locales between the
11041 *ctags* file creator and the *vi* file reader.

11042 Historically, the tags file has been used only by *ex* and *vi*. However, the format of the tags file
11043 has been published to encourage other programs to use the tags in new ways. The format allows
11044 either patterns or line numbers to find the identifiers because the historical *vi* recognizes either.
11045 The *ctags* utility does not produce the format using line numbers because it is not useful
11046 following any source file changes that add or delete lines. The documented search patterns
11047 match historical practice. It should be noted that literal leading circumflex or trailing dollar-sign
11048 characters in the search pattern will only behave correctly if anchored to the beginning of the
11049 line or end of the line by an additional circumflex or dollar-sign character.

11050 Historical implementations also understand the objects used by the languages Pascal and
11051 sometimes LISP, and they understand the C source output by *lex* and *yacc*. The *ctags* utility is
11052 not required to accommodate these languages, although implementors are encouraged to do so.

11053 The following historical option was not specified, as *vgrind* is not included in this volume of
11054 IEEE Std 1003.1-2001:

11055 **-v** If the **-v** flag is given, an index of the form expected by *vgrind* is produced on the
11056 standard output. This listing contains the function name, filename, and page
11057 number (assuming 64-line pages). Since the output is sorted into lexicographic
11058 order, it may be desired to run the output through *sort -f*. Sample use:

11059 `ctags -v files | sort -f > index vgrind -x index`

11060 The special treatment of the tag **main** makes the use of *ctags* practical in directories with more
11061 than one program.

11062 FUTURE DIRECTIONS

11063 None.

11064 SEE ALSO

11065 *c99*, *fort77*, *vi*

11066 CHANGE HISTORY

11067 First released in Issue 4.

11068 Issue 5

11069 The FUTURE DIRECTIONS section is added.

11070 Issue 6

11071 This utility is marked as part of the User Portability Utilities option.

11072 The OUTPUT FILES section is changed to align with the IEEE P1003.2b draft standard.

11073 The normative text is reworded to avoid use of the term “must” for application requirements.

11074 IEEE PASC Interpretation 1003.2 #168 is applied, changing “create” to “write” in the
11075 DESCRIPTION.

11076 NAME

11077 cut — cut out selected fields of each line of a file

11078 SYNOPSIS

11079 cut -b *list* [-n] [*file* ...]

11080 cut -c *list* [*file* ...]

11081 cut -f *list* [-d *delim*] [-s] [*file* ...]

11082 DESCRIPTION

11083 The *cut* utility shall cut out bytes (-b option), characters (-c option), or character-delimited fields
11084 (-f option) from each line in one or more files, concatenate them, and write them to standard
11085 output.

11086 OPTIONS

11087 The *cut* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
11088 12.2, Utility Syntax Guidelines.

11089 The application shall ensure that the option-argument *list* (see options -b, -c, and -f below) is a
11090 comma-separated list or <blank>-separated list of positive numbers and ranges. Ranges can be
11091 in three forms. The first is two positive numbers separated by a hyphen (*low-high*), which
11092 represents all fields from the first number to the second number. The second is a positive
11093 number preceded by a hyphen (-*high*), which represents all fields from field number 1 to that
11094 number. The third is a positive number followed by a hyphen (*low-*), which represents that
11095 number to the last field, inclusive. The elements in *list* can be repeated, can overlap, and can be
11096 specified in any order, but the bytes, characters, or fields selected shall be written in the order of
11097 the input data. If an element appears in the selection list more than once, it shall be written
11098 exactly once.

11099 The following options shall be supported:

11100 **-b** *list* Cut based on a *list* of bytes. Each selected byte shall be output unless the -n option
11101 is also specified. It shall not be an error to select bytes not present in the input line.

11102 **-c** *list* Cut based on a *list* of characters. Each selected character shall be output. It shall
11103 not be an error to select characters not present in the input line.

11104 **-d** *delim* Set the field delimiter to the character *delim*. The default is the <tab>.

11105 **-f** *list* Cut based on a *list* of fields, assumed to be separated in the file by a delimiter
11106 character (see -d). Each selected field shall be output. Output fields shall be
11107 separated by a single occurrence of the field delimiter character. Lines with no field
11108 delimiters shall be passed through intact, unless -s is specified. It shall not be an
11109 error to select fields not present in the input line.

11110 **-n** Do not split characters. When specified with the -b option, each element in *list* of
11111 the form *low-high* (hyphen-separated numbers) shall be modified as follows:

- 11112 • If the byte selected by *low* is not the first byte of a character, *low* shall be
11113 decremented to select the first byte of the character originally selected by *low*.
11114 If the byte selected by *high* is not the last byte of a character, *high* shall be
11115 decremented to select the last byte of the character prior to the character
11116 originally selected by *high*, or zero if there is no prior character. If the resulting
11117 range element has *high* equal to zero or *low* greater than *high*, the list element
11118 shall be dropped from *list* for that input line without causing an error.

11119 Each element in *list* of the form *low-* shall be treated as above with *high* set to the
11120 number of bytes in the current line, not including the terminating <newline>. Each

11121 element in *list* of the form *-high* shall be treated as above with *low* set to 1. Each
11122 element in *list* of the form *num* (a single number) shall be treated as above with *low*
11123 set to *num* and *high* set to *num*.

11124 **-s** Suppress lines with no delimiter characters, when used with the **-f** option. Unless
11125 specified, lines with no delimiters shall be passed through untouched.

11126 OPERANDS

11127 The following operand shall be supported:

11128 *file* A pathname of an input file. If no *file* operands are specified, or if a *file* operand is
11129 '*-*', the standard input shall be used.

11130 STDIN

11131 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '*-*'.
11132 See the INPUT FILES section.

11133 INPUT FILES

11134 The input files shall be text files, except that line lengths shall be unlimited.

11135 ENVIRONMENT VARIABLES

11136 The following environment variables shall affect the execution of *cut*:

11137 *LANG* Provide a default value for the internationalization variables that are unset or null.
11138 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
11139 Internationalization Variables for the precedence of internationalization variables
11140 used to determine the values of locale categories.)

11141 *LC_ALL* If set to a non-empty string value, override the values of all the other
11142 internationalization variables.

11143 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
11144 characters (for example, single-byte as opposed to multi-byte characters in
11145 arguments and input files).

11146 *LC_MESSAGES*

11147 Determine the locale that should be used to affect the format and contents of
11148 diagnostic messages written to standard error.

11149 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

11150 ASYNCHRONOUS EVENTS

11151 Default.

11152 STDOUT

11153 The *cut* utility output shall be a concatenation of the selected bytes, characters, or fields (one of
11154 the following):

11155 "%s\n", <concatenation of bytes>

11156 "%s\n", <concatenation of characters>

11157 "%s\n", <concatenation of fields and field delimiters>

11158 STDERR

11159 The standard error shall be used only for diagnostic messages.

11160 OUTPUT FILES

11161 None.

11162 EXTENDED DESCRIPTION

11163 None.

11164 EXIT STATUS

11165 The following exit values shall be returned:

11166 0 All input files were output successfully.

11167 >0 An error occurred.

11168 CONSEQUENCES OF ERRORS

11169 Default.

11170 APPLICATION USAGE

11171 Earlier versions of the *cut* utility worked in an environment where bytes and characters were
11172 considered equivalent (modulo <backspace> and <tab> processing in some implementations). In
11173 the extended world of multi-byte characters, the new **-b** option has been added. The **-n** option
11174 (used with **-b**) allows it to be used to act on bytes rounded to character boundaries. The
11175 algorithm specified for **-n** guarantees that:

11176 `cut -b 1-500 -n file > file1`11177 `cut -b 501- -n file > file2`

11178 ends up with all the characters in **file** appearing exactly once in **file1** or **file2**. (There is,
11179 however, a <newline> in both **file1** and **file2** for each <newline> in **file**.)

11180 EXAMPLES

11181 Examples of the option qualifier list:

11182 1,4,7 Select the first, fourth, and seventh bytes, characters, or fields and field delimiters.

11183 1-3,8 Equivalent to 1,2,3,8.

11184 -5,10 Equivalent to 1,2,3,4,5,10.

11185 3- Equivalent to third to last, inclusive.

11186 The *low-high* forms are not always equivalent when used with **-b** and **-n** and multi-byte
11187 characters; see the description of **-n**.

11188 The following command:

11189 `cut -d : -f 1,6 /etc/passwd`

11190 reads the System V password file (user database) and produces lines of the form:

11191 `<user ID>:<home directory>`

11192 Most utilities in this volume of IEEE Std 1003.1-2001 work on text files. The *cut* utility can be
11193 used to turn files with arbitrary line lengths into a set of text files containing the same data. The
11194 *paste* utility can be used to create (or recreate) files with arbitrary line lengths. For example, if **file**
11195 contains long lines:

11196 `cut -b 1-500 -n file > file1`11197 `cut -b 501- -n file > file2`

11198 creates **file1** (a text file) with lines no longer than 500 bytes (plus the <newline>) and **file2** that
11199 contains the remainder of the data from **file**. (Note that **file2** is not a text file if there are lines in
11200 **file** that are longer than 500 + {LINE_MAX} bytes.) The original file can be recreated from **file1**
11201 and **file2** using the command:

11202 `paste -d "\0" file1 file2 > file`

RATIONALE

Some historical implementations do not count `<backspace>`s in determining character counts with the `-c` option. This may be useful for using *cut* for processing *nroff* output. It was deliberately decided not to have the `-c` option treat either `<backspace>`s or `<tab>`s in any special fashion. The *fold* utility does treat these characters specially.

Unlike other utilities, some historical implementations of *cut* exit after not finding an input file, rather than continuing to process the remaining *file* operands. This behavior is prohibited by this volume of IEEE Std 1003.1-2001, where only the exit status is affected by this problem.

The behavior of *cut* when provided with either mutually-exclusive options or options that do not work logically together has been deliberately left unspecified in favor of global wording in Section 1.11 (on page 20).

The OPTIONS section was changed in response to IEEE PASC Interpretation 1003.2 #149. The change represents historical practice on all known systems. The original standard was ambiguous on the nature of the output.

The *list* option-arguments are historically used to select the portions of the line to be written, but do not affect the order of the data. For example:

```
echo abcdefghi | cut -c6,2,4-7,1
```

yields "abdefg".

A proposal to enhance *cut* with the following option:

-o Preserve the selected field order. When this option is specified, each byte, character, or field (or ranges of such) shall be written in the order specified by the *list* option-argument, even if this requires multiple outputs of the same bytes, characters, or fields.

was rejected because this type of enhancement is outside the scope of the IEEE P1003.2b draft standard.

FUTURE DIRECTIONS

None.

SEE ALSO

grep, *paste*, Section 2.5 (on page 33)

CHANGE HISTORY

First released in Issue 2.

Issue 6

The OPTIONS section is changed to align with the IEEE P1003.2b draft standard.

The normative text is reworded to avoid use of the term “must” for application requirements.

11236 NAME

11237 cxref — generate a C-language program cross-reference table (**DEVELOPMENT**)

11238 SYNOPSIS

```
11239 XSI      cxref [-cs][-o file][-w num] [-D name[=def]]...[-I dir]...
11240           [-U name]... file ...
```

11241

11242 DESCRIPTION

11243 The *cxref* utility shall analyze a collection of C-language *files* and attempt to build a cross-
 11244 reference table. Information from **#define** lines shall be included in the symbol table. A sorted
 11245 listing shall be written to standard output of all symbols (auto, static, and global) in each *file*
 11246 separately, or with the **-c** option, in combination. Each symbol shall contain an asterisk before
 11247 the declaring reference.

11248 OPTIONS

11249 The *cxref* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
 11250 12.2, Utility Syntax Guidelines, except that the order of the **-D**, **-I**, and **-U** options (which are
 11251 identical to their interpretation by *c99*) is significant. The following options shall be supported:

- 11252 **-c** Write a combined cross-reference of all input files.
- 11253 **-s** Operate silently; do not print input filenames.
- 11254 **-o** *file* Direct output to named *file*.
- 11255 **-w** *num* Format output no wider than *num* (decimal) columns. This option defaults to 80 if
 11256 *num* is not specified or is less than 51.
- 11257 **-D** Equivalent to *c99*.
- 11258 **-I** Equivalent to *c99*.
- 11259 **-U** Equivalent to *c99*.

11260 OPERANDS

11261 The following operand shall be supported:

11262 *file* A pathname of a C-language source file.

11263 STDIN

11264 Not used.

11265 INPUT FILES

11266 The input files are C-language source files.

11267 ENVIRONMENT VARIABLES

11268 The following environment variables shall affect the execution of *cxref*:

- 11269 **LANG** Provide a default value for the internationalization variables that are unset or null.
 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
 Internationalization Variables for the precedence of internationalization variables
 used to determine the values of locale categories.)
- 11273 **LC_ALL** If set to a non-empty string value, override the values of all the other
 internationalization variables.
- 11275 **LC_COLLATE** Determine the locale for the ordering of the output.
- 11277 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
 characters (for example, single-byte as opposed to multi-byte characters in

- 11279 arguments and input files).
- 11280 **LC_MESSAGES**
- 11281 Determine the locale that should be used to affect the format and contents of
- 11282 diagnostic messages written to standard error.
- 11283 **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.
- 11284 **ASYNCHRONOUS EVENTS**
- 11285 Default.
- 11286 **STDOUT**
- 11287 The standard output shall be used for the cross-reference listing, unless the **-o** option is used to
- 11288 select a different output file.
- 11289 The format of standard output is unspecified, except that the following information shall be
- 11290 included:
- 11291 • If the **-c** option is not specified, each portion of the listing shall start with the name of the
 - 11292 input file on a separate line.
 - 11293 • The name line shall be followed by a sorted list of symbols, each with its associated location
 - 11294 pathname, the name of the function in which it appears (if it is not a function name itself),
 - 11295 and line number references.
 - 11296 • Each line number may be preceded by an asterisk ('*') flag, meaning that this is the
 - 11297 declaring reference. Other single-character flags, with implementation-defined meanings,
 - 11298 may be included.
- 11299 **STDERR**
- 11300 The standard error shall be used only for diagnostic messages.
- 11301 **OUTPUT FILES**
- 11302 The output file named by the **-o** option shall be used instead of standard output.
- 11303 **EXTENDED DESCRIPTION**
- 11304 None.
- 11305 **EXIT STATUS**
- 11306 The following exit values shall be returned:
- 11307 0 Successful completion.
- 11308 >0 An error occurred.
- 11309 **CONSEQUENCES OF ERRORS**
- 11310 Default.
- 11311 **APPLICATION USAGE**
- 11312 None.
- 11313 **EXAMPLES**
- 11314 None.
- 11315 **RATIONALE**
- 11316 None.
- 11317 **FUTURE DIRECTIONS**
- 11318 None.

11319 **SEE ALSO**

11320 *c99*

11321 **CHANGE HISTORY**

11322 First released in Issue 2.

11323 **Issue 5**

11324 In the SYNOPSIS, **[−U dir]** is changed to **[−U name]**.

11325 **Issue 6**

11326 The APPLICATION USAGE section is added.

11327 NAME

11328 date — write the date and time

11329 SYNOPSIS

11330 date [-u] [+format]

11331 XSI date [-u] mmddhhmm[[cc]YY]

11332

11333 DESCRIPTION

11334 XSI The *date* utility shall write the date and time to standard output or attempt to set the system date and time. By default, the current date and time shall be written. If an operand beginning with '+' is specified, the output format of *date* shall be controlled by the conversion specifications and other text in the operand.

11338 OPTIONS

11339 The *date* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

11341 The following option shall be supported:

11342 **-u** Perform operations as if the *TZ* environment variable was set to the string "UTC0", or its equivalent historical value of "GMT0". Otherwise, *date* shall use the timezone indicated by the *TZ* environment variable or the system default if that variable is unset or null.

11346 OPERANDS

11347 The following operands shall be supported:

11348 **+format** When the format is specified, each conversion specifier shall be replaced in the standard output by its corresponding value. All other characters shall be copied to the output without change. The output shall always be terminated with a <newline>.

11352 Conversion Specifications

11353 %a	Locale's abbreviated weekday name.
11354 %A	Locale's full weekday name.
11355 %b	Locale's abbreviated month name.
11356 %B	Locale's full month name.
11357 %c	Locale's appropriate date and time representation.
11358 %C	Century (a year divided by 100 and truncated to an integer) as a decimal number [00,99].
11360 %d	Day of the month as a decimal number [01,31].
11361 %D	Date in the format <i>mm/dd/yy</i> .
11362 %e	Day of the month as a decimal number [1,31] in a two-digit field with leading space character fill.
11364 %h	A synonym for %b.
11365 %H	Hour (24-hour clock) as a decimal number [00,23].
11366 %I	Hour (12-hour clock) as a decimal number [01,12].

11367	%j	Day of the year as a decimal number [001,366].
11368	%m	Month as a decimal number [01,12].
11369	%M	Minute as a decimal number [00,59].
11370	%n	A <newline>.
11371	%p	Locale's equivalent of either AM or PM.
11372	%r	12-hour clock time [01,12] using the AM/PM notation; in the POSIX locale, this shall be equivalent to %I:%M:%S %p.
11374	%S	Seconds as a decimal number [00,60].
11375	%t	A <tab>.
11376	%T	24-hour clock time [00,23] in the format HH:MM:SS.
11377	%u	Weekday as a decimal number [1,7] (1=Monday).
11378	%U	Week of the year (Sunday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Sunday shall be considered to be in week 0.
11381	%V	Week of the year (Monday as the first day of the week) as a decimal number [01,53]. If the week containing January 1 has four or more days in the new year, then it shall be considered week 1; otherwise, it shall be the last week of the previous year, and the next week shall be week 1.
11385	%w	Weekday as a decimal number [0,6] (0=Sunday).
11386	%W	Week of the year (Monday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Monday shall be considered to be in week 0.
11389	%x	Locale's appropriate date representation.
11390	%X	Locale's appropriate time representation.
11391	%Y	Year within century [00,99].
11392	%Y	Year with century as a decimal number.
11393	%Z	Timezone name, or no characters if no timezone is determinable.
11394	%%	A percent sign character.
11395		See the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3.5, LC_TIME for the conversion specifier values in the POSIX locale.
11396		

11397 Modified Conversion Specifications

Some conversion specifiers can be modified by the `E` and `O` modifier characters to indicate a different format or specification as specified in the `LC_TIME` locale description (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3.5, `LC_TIME`). If the corresponding keyword (see `era`, `era_year`, `era_d_fmt`, and `alt_digits` in the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3.5, `LC_TIME`) is not specified or not supported for the current locale, the unmodified conversion specifier value shall be used.

11405 %Ec Locale's alternative appropriate date and time representation.

11406	%EC	The name of the base year (period) in the locale's alternative representation.
11407		
11408	%Ex	Locale's alternative date representation.
11409	%Ex	Locale's alternative time representation.
11410	%Ey	Offset from %EC (year only) in the locale's alternative representation.
11411	%EY	Full alternative year representation.
11412	%Od	Day of month using the locale's alternative numeric symbols.
11413	%Oe	Day of month using the locale's alternative numeric symbols.
11414	%OH	Hour (24-hour clock) using the locale's alternative numeric symbols.
11415	%OI	Hour (12-hour clock) using the locale's alternative numeric symbols.
11416	%Om	Month using the locale's alternative numeric symbols.
11417	%OM	Minutes using the locale's alternative numeric symbols.
11418	%OS	Seconds using the locale's alternative numeric symbols.
11419	%Ou	Weekday as a number in the locale's alternative representation (Monday = 1).
11420		
11421	%OU	Week number of the year (Sunday as the first day of the week) using the locale's alternative numeric symbols.
11422		
11423	%OV	Week number of the year (Monday as the first day of the week, rules corresponding to %v), using the locale's alternative numeric symbols.
11424		
11425	%Ow	Weekday as a number in the locale's alternative representation (Sunday = 0).
11426		
11427	%OW	Week number of the year (Monday as the first day of the week) using the locale's alternative numeric symbols.
11428		
11429	%Oy	Year (offset from %C) in alternative representation.
11430 XSI	mddhhmm[[cc]yy]	Attempt to set the system date and time from the value given in the operand. This is only possible if the user has appropriate privileges and the system permits the setting of the system date and time. The first <i>mm</i> is the month (number); <i>dd</i> is the day (number); <i>hh</i> is the hour (number, 24-hour system); the second <i>mm</i> is the minute (number); <i>cc</i> is the century and is the first two digits of the year (this is optional); <i>yy</i> is the last two digits of the year and is optional. If century is not specified, then values in the range [69,99] shall refer to years 1969 to 1999 inclusive, and values in the range [00,68] shall refer to years 2000 to 2068 inclusive. The current year is the default if <i>yy</i> is omitted.
11440	Note:	It is expected that in a future version of IEEE Std 1003.1-2001 the default century inferred from a 2-digit year will change. (This would apply to all commands accepting a 2-digit year as input.)
11441		
11442		
11443 STDIN		
11444	Not used.	

11445 INPUT FILES

11446 None.

11447 ENVIRONMENT VARIABLES

11448 The following environment variables shall affect the execution of *date*:

11449 *LANG* Provide a default value for the internationalization variables that are unset or null.
11450 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
11451 Internationalization Variables for the precedence of internationalization variables
11452 used to determine the values of locale categories.)

11453 *LC_ALL* If set to a non-empty string value, override the values of all the other
11454 internationalization variables.

11455 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
11456 characters (for example, single-byte as opposed to multi-byte characters in
11457 arguments).

11458 *LC_MESSAGES*

11459 Determine the locale that should be used to affect the format and contents of
11460 diagnostic messages written to standard error.

11461 *LC_TIME* Determine the format and contents of date and time strings written by *date*.

11462 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

11463 *TZ* Determine the timezone in which the time and date are written, unless the **-u**
11464 option is specified. If the *TZ* variable is unset or null and **-u** is not specified, an
11465 unspecified system default timezone is used.

11466 ASYNCHRONOUS EVENTS

11467 Default.

11468 STDOUT

11469 When no formatting operand is specified, the output in the POSIX locale shall be equivalent to
11470 specifying:

11471 `date "+%a %b %e %H:%M:%S %Z %Y"`

11472 STDERR

11473 The standard error shall be used only for diagnostic messages.

11474 OUTPUT FILES

11475 None.

11476 EXTENDED DESCRIPTION

11477 None.

11478 EXIT STATUS

11479 The following exit values shall be returned:

11480 0 The date was written successfully.

11481 >0 An error occurred.

11482 CONSEQUENCES OF ERRORS

11483 Default.

11484 APPLICATION USAGE

11485 Conversion specifiers are of unspecified format when not in the POSIX locale. Some of them can
 11486 contain <newline>s in some locales, so it may be difficult to use the format shown in standard
 11487 output for parsing the output of *date* in those locales.

11488 The range of values for %S extends from 0 to 60 seconds to accommodate the occasional leap
 11489 second.

11490 Although certain of the conversion specifiers in the POSIX locale (such as the name of the
 11491 month) are shown with initial capital letters, this need not be the case in other locales. Programs
 11492 using these fields may need to adjust the capitalization if the output is going to be used at the
 11493 beginning of a sentence.

11494 The date string formatting capabilities are intended for use in Gregorian-style calendars,
 11495 possibly with a different starting year (or years). The %x and %c conversion specifications,
 11496 however, are intended for local representation; these may be based on a different, non-Gregorian
 11497 calendar.

11498 The %C conversion specification was introduced to allow a fallback for the %EC (alternative year
 11499 format base year); it can be viewed as the base of the current subdivision in the Gregorian
 11500 calendar. The century number is calculated as the year divided by 100 and truncated to an
 11501 integer; it should not be confused with the use of ordinal numbers for centuries (for example,
 11502 "twenty-first century"). Both the %EY and %Y can then be viewed as the offset from %EC and %C,
 11503 respectively.

11504 The E and O modifiers modify the traditional conversion specifiers, so that they can always be
 11505 used, even if the implementation (or the current locale) does not support the modifier.

11506 The E modifier supports alternative date formats, such as the Japanese Emperor's Era, as long as
 11507 these are based on the Gregorian calendar system. Extending the E modifiers to other date
 11508 elements may provide an implementation-defined extension capable of supporting other
 11509 calendar systems, especially in combination with the O modifier.

11510 The O modifier supports time and date formats using the locale's alternative numerical symbols,
 11511 such as Kanji or Hindi digits or ordinal number representation.

11512 Non-European locales, whether they use Latin digits in computational items or not, often have
 11513 local forms of the digits for use in date formats. This is not totally unknown even in Europe; a
 11514 variant of dates uses Roman numerals for the months: the third day of September 1991 would be
 11515 written as 3.IX.1991. In Japan, Kanji digits are regularly used for dates; in Arabic-speaking
 11516 countries, Hindi digits are used. The %d, %e, %H, %I, %m, %S, %U, %w, %W, and %Y conversion
 11517 specifications always return the date and time field in Latin digits (that is, 0 to 9). The %O
 11518 modifier was introduced to support the use for display purposes of non-Latin digits. In the
 11519 *LC_TIME* category in *localedef*, the optional **alt_digits** keyword is intended for this purpose. As
 11520 an example, assume the following (partial) *localedef* source:

```
11521 alt_digits    " " ; "I" ; "II" ; "III" ; "IV" ; "V" ; "VI" ; "VII" ; "VIII" \ 
11522           "IX" ; "X" ; "XI" ; "XII"
11523 d_fmt        "%e.%Om.%Y"
```

11524 With the above date, the command:

```
11525 date "+%x"
```

11526 would yield 3.IX.1991. With the same **d_fmt**, but without the **alt_digits**, the command would
 11527 yield 3.9.1991.

11528 EXAMPLES

- 11529 1. The following are input/output examples of *date* used at arbitrary times in the POSIX
 11530 locale:

```
11531 $ date
11532 Tue Jun 26 09:58:10 PDT 1990
11533 $ date "+DATE: %m/%d/%Y%nTIME: %H:%M:%S"
11534 DATE: 11/02/91
11535 TIME: 13:36:16
11536 $ date "+TIME: %r"
11537 TIME: 01:36:32 PM
```

- 11538 2. Examples for Denmark, where the default date and time format is %a %d %b %Y %T %Z:

```
11539 $ LANG=da_DK.iso_8859-1 date
11540 ons 02 okt 1991 15:03:32 CET
11541 $ LANG=da_DK.iso_8859-1 \
11542     date "+DATO: %A den %e. %B %Y%nKLOKKEN: %H:%M:%S"
11543 DATO: onsdag den 2. oktober 1991
11544 KLOKKEN: 15:03:56
```

- 11545 3. Examples for Germany, where the default date and time format is %a %d.%h.%Y, %T %Z:

```
11546 $ LANG=De_DE.88591 date
11547 Mi 02.Okt.1991, 15:01:21 MEZ
11548 $ LANG=De_DE.88591 date "+DATUM: %A, %d. %B %Y%nZEIT: %H:%M:%S"
11549 DATUM: Mittwoch, 02. Oktober 1991
11550 ZEIT: 15:02:02
```

- 11551 4. Examples for France, where the default date and time format is %a %d %h %Y %Z %T:

```
11552 $ LANG=Fr_FR.88591 date
11553 Mer 02 oct 1991 MET 15:03:32
11554 $ LANG=Fr_FR.88591 date "+JOUR: %A %d %B %Y%nHEURE: %H:%M:%S"
11555 JOUR: Mercredi 02 octobre 1991
11556 HEURE: 15:03:56
```

11557 RATIONALE

11558 Some of the new options for formatting are from the ISO C standard. The **-u** option was
 11559 introduced to allow portable access to Coordinated Universal Time (UTC). The string "GMT0" is
 11560 allowed as an equivalent *TZ* value to be compatible with all of the systems using the BSD
 11561 implementation, where this option originated.

11562 The %e format conversion specification (adopted from System V) was added because the ISO C
 11563 standard conversion specifications did not provide any way to produce the historical default
 11564 *date* output during the first nine days of any month.

11565 There are two varieties of day and week numbering supported (in addition to any others created
 11566 with the locale-dependent %E and %O modifier characters):

- 11567 • The historical variety in which Sunday is the first day of the week and the weekdays
 11568 preceding the first Sunday of the year are considered week 0. These are represented by %w
 11569 and %U. A variant of this is %w, using Monday as the first day of the week, but still referring
 11570 to week 0. This view of the calendar was retained because so many historical applications
 11571 depend on it and the ISO C standard *strftime()* function, on which many *date*

- 11572 implementations are based, was defined in this way.
- 11573 • The international standard, based on the ISO 8601: 2000 standard where Monday is the first
11574 weekday and the algorithm for the first week number is more complex: If the week (Monday
11575 to Sunday) containing January 1 has four or more days in the new year, then it is week 1;
11576 otherwise, it is week 53 of the previous year, and the next week is week 1. These are
11577 represented by the new conversion specifications %u and %v, added as a result of
11578 international comments.
- 11579 **FUTURE DIRECTIONS**
- 11580 None.
- 11581 **SEE ALSO**
- 11582 The System Interfaces volume of IEEE Std 1003.1-2001, *printf()*, *strftime()*
- 11583 **CHANGE HISTORY**
- 11584 First released in Issue 2.
- 11585 **Issue 5**
- 11586 Changes are made for Year 2000 alignment.
- 11587 **Issue 6**
- 11588 The following new requirements on POSIX implementations derive from alignment with the
11589 Single UNIX Specification:
- 11590 • The setting of system date and time is described, including how to interpret two-digit year
11591 values if a century is not given.
- 11592 • The %EX modified conversion specification is added.
- 11593 The Open Group Corrigendum U048/2 is applied, correcting the examples.
- 11594 The DESCRIPTION is updated to refer to conversion specifications, instead of field descriptors
11595 for consistency with the *LC_TIME* category.
- 11596 A clarification is made such that the current year is the default if the yy argument is omitted
11597 when setting the system date and time.

11598 NAME

11599 dd — convert and copy a file

11600 SYNOPSIS

11601 dd [*operand* . . .]

11602 DESCRIPTION

11603 The **dd** utility shall copy the specified input file to the specified output file with possible
11604 conversions using specific input and output block sizes. It shall read the input one block at a
11605 time, using the specified input block size; it shall then process the block of data actually
11606 returned, which could be smaller than the requested block size. It shall apply any conversions
11607 that have been specified and write the resulting data to the output in blocks of the specified
11608 output block size. If the **bs=expr** operand is specified and no conversions other than **sync**,
11609 **noerror**, or **notrunc** are requested, the data returned from each input block shall be written as a
11610 separate output block; if the read returns less than a full block and the **sync** conversion is not
11611 specified, the resulting output block shall be the same size as the input block. If the **bs=expr**
11612 operand is not specified, or a conversion other than **sync**, **noerror**, or **notrunc** is requested, the
11613 input shall be processed and collected into full-sized output blocks until the end of the input is
11614 reached.

11615 The processing order shall be as follows:

- 11616 1. An input block is read.
- 11617 2. If the input block is shorter than the specified input block size and the **sync** conversion is
11618 specified, null bytes shall be appended to the input data up to the specified size. (If either
11619 **block** or **unblock** is also specified, <space>s shall be appended instead of null bytes.) The
11620 remaining conversions and output shall include the pad characters as if they had been read
11621 from the input.
- 11622 3. If the **bs=expr** operand is specified and no conversion other than **sync** or **noerror** is
11623 requested, the resulting data shall be written to the output as a single block, and the
11624 remaining steps are omitted.
- 11625 4. If the **swab** conversion is specified, each pair of input data bytes shall be swapped. If there
11626 is an odd number of bytes in the input block, the last byte in the input record shall not be
11627 swapped.
- 11628 5. Any remaining conversions (**block**, **unblock**, **lcase**, and **ucase**) shall be performed. These
11629 conversions shall operate on the input data independently of the input blocking; an input
11630 or output fixed-length record may span block boundaries.
- 11631 6. The data resulting from input or conversion or both shall be aggregated into output blocks
11632 of the specified size. After the end of input is reached, any remaining output shall be
11633 written as a block without padding if **conv=sync** is not specified; thus, the final output
11634 block may be shorter than the output block size.

11635 OPTIONS

11636 None.

11637 OPERANDS

11638 All of the operands shall be processed before any input is read. The following operands shall be
11639 supported:

- | | |
|------------------------------|---|
| 11640 if= <i>file</i> | Specify the input pathname; the default is standard input. |
| 11641 of= <i>file</i> | Specify the output pathname; the default is standard output. If the seek=expr
11642 conversion is not also specified, the output file shall be truncated before the copy
11643 begins if an explicit of= <i>file</i> operand is specified, unless conv=notrunc is specified. |

11644	If seek=expr is specified, but conv=notrunc is not, the effect of the copy shall be to preserve the blocks in the output file over which dd seeks, but no other portion of the output file shall be preserved. (If the size of the seek plus the size of the input file is less than the previous size of the output file, the output file shall be shortened by the copy.)	
11649	ibs=expr	Specify the input block size, in bytes, by <i>expr</i> (default is 512).
11650	obs=expr	Specify the output block size, in bytes, by <i>expr</i> (default is 512).
11651	bs=expr	Set both input and output block sizes to <i>expr</i> bytes, superseding ibs= and obs= . If no conversion other than sync , noerror , and notrunc is specified, each input block shall be copied to the output as a single block without aggregating short blocks.
11654	cbs=expr	Specify the conversion block size for block and unblock in bytes by <i>expr</i> (default is zero). If cbs= is omitted or given a value of zero, using block or unblock produces unspecified results.
11657 XSI	The application shall ensure that this operand is also specified if the conv= operand is specified with a value of ascii , ebcdic , or ibm . For a conv= operand with an ascii value, the input is handled as described for the unblock value, except that characters are converted to ASCII before any trailing <space>s are deleted. For conv= operands with ebcdic or ibm values, the input is handled as described for the block value except that the characters are converted to EBCDIC or IBM EBCDIC, respectively, after any trailing <space>s are added.	
11664	skip=n	Skip <i>n</i> input blocks (using the specified input block size) before starting to copy. On seekable files, the implementation shall read the blocks or seek past them; on non-seekable files, the blocks shall be read and the data shall be discarded.
11667	seek=n	Skip <i>n</i> blocks (using the specified output block size) from the beginning of the output file before copying. On non-seekable files, existing blocks shall be read and space from the current end-of-file to the specified offset, if any, filled with null bytes; on seekable files, the implementation shall seek to the specified offset or read the blocks as described for non-seekable files.
11672	count=n	Copy only <i>n</i> input blocks.
11673	conv=value[,value...] Where <i>values</i> are comma-separated symbols from the following list:	
11675 XSI	ascii	Convert EBCDIC to ASCII; see Table 4-6 (on page 303).
11676 XSI	ebcdic	Convert ASCII to EBCDIC; see Table 4-6 (on page 303).
11677 XSI	ibm	Convert ASCII to a different EBCDIC set; see Table 4-7 (on page 304).
11678	The ascii , ebcdic , and ibm values are mutually-exclusive.	
11679	block	Treat the input as a sequence of <newline>-terminated or end-of-file-terminated variable-length records independent of the input block boundaries. Each record shall be converted to a record with a fixed length specified by the conversion block size. Any <newline> shall be removed from the input line; <space>s shall be appended to lines that are shorter than their conversion block size to fill the block. Lines that are longer than the conversion block size shall be truncated to the largest number of characters that fit into that size; the number of truncated lines shall be reported (see the STDERR section).

11689		The block and unblock values are mutually-exclusive.
11690	unblock	Convert fixed-length records to variable length. Read a number of bytes equal to the conversion block size (or the number of bytes remaining in the input, if less than the conversion block size), delete all trailing <space>s, and append a <newline>.
11691		
11692		
11693		
11694	lcase	Map uppercase characters specified by the <i>LC_CTYPE</i> keyword tolower to the corresponding lowercase character. Characters for which no mapping is specified shall not be modified by this conversion.
11695		
11696		
11697		
11698		The lcase and ucase symbols are mutually-exclusive.
11699	ucase	Map lowercase characters specified by the <i>LC_CTYPE</i> keyword toupper to the corresponding uppercase character. Characters for which no mapping is specified shall not be modified by this conversion.
11700		
11701		
11702		
11703	swab	Swap every pair of input bytes.
11704	noerror	Do not stop processing on an input error. When an input error occurs, a diagnostic message shall be written on standard error, followed by the current input and output block counts in the same format as used at completion (see the STDERR section). If the sync conversion is specified, the missing input shall be replaced with null bytes and processed normally; otherwise, the input block shall be omitted from the output.
11705		
11706		
11707		
11708		
11709		
11710		
11711	notrunc	Do not truncate the output file. Preserve blocks in the output file not explicitly written by this invocation of the <i>dd</i> utility. (See also the preceding of= <i>file</i> operand.)
11712		
11713		
11714	sync	Pad every input block to the size of the ibs= buffer, appending null bytes. (If either block or unblock is also specified, append <space>s, rather than null bytes.)
11715		
11716		
11717		The behavior is unspecified if operands other than conv= are specified more than once.
11718		For the bs= , cbs= , ibs= , and obs= operands, the application shall supply an expression specifying a size in bytes. The expression, <i>expr</i> , can be:
11719		
11720	1.	A positive decimal number
11721	2.	A positive decimal number followed by <i>k</i> , specifying multiplication by 1 024
11722	3.	A positive decimal number followed by <i>b</i> , specifying multiplication by 512
11723	4.	Two or more positive decimal numbers (with or without <i>k</i> or <i>b</i>) separated by <i>x</i> , specifying the product of the indicated values
11724		
11725		All of the operands are processed before any input is read.
11726	XSI	The following two tables display the octal number character values used for the ascii and ebcdic conversions (first table) and for the ibm conversion (second table). In both tables, the ASCII values are the row and column headers and the EBCDIC values are found at their intersections. For example, ASCII 0012 (LF) is the second row, third column, yielding 0045 in EBCDIC. The inverted tables (for EBCDIC to ASCII conversion) are not shown, but are in one-to-one correspondence with these tables. The differences between the two tables are highlighted by small boxes drawn around five entries.
11727		
11728		
11729		
11730		
11731		
11732		

Table 4-6 ASCII to EBCDIC Conversion

	0	1	2	3	4	5	6	7
0000	0000 NUL	0001 SOH	0002 STX	0003 ETX	0067 EOT	0055 ENQ	0056 ACK	0057 BEL
0010	0026 BS	0005 HT	0045 LF	0013 VT	0014 FF	0015 CR	0016 SO	0017 SI
0020	0020 DLE	0021 DC1	0022 DC2	0023 DC3	0074 DC4	0075 NAK	0062 SYN	0046 ETB
0030	0030 CAN	0031 EM	0077 SUB	0047 ESC	0034 IFS	0035 IGS	0036 IRS	0037 ITB
0040	0100 Sp	0132 !	0177 "	0173 #	0133 \$	0154 %	0120 &	0175 '
0050	0115 (0135)	0134 *	0116 +	0153 ,	0140 -	0113 .	0141 /
0060	0360 0	0361 1	0362 2	0363 3	0364 4	0365 5	0366 6	0367 7
0070	0370 8	0371 9	0172 :	0136 ;	0114 <	0176 =	0156 >	0157 ?
0100	0174 @	0301 A	0302 B	0303 C	0304 D	0305 E	0306 F	0307 G
0110	0310 H	0311 I	0321 J	0322 K	0323 L	0324 M	0325 N	0326 O
0120	0327 P	0330 Q	0331 R	0342 S	0343 T	0344 U	0345 V	0346 W
0130	0347 X	0350 Y	0351 Z	0255 [0340 \	0275]	0232	0155 _
0140	0171 `	0201 a	0202 b	0203 c	0204 d	0205 e	0206 f	0207 g
0150	0210 h	0211 i	0221 j	0222 k	0223]	0224 m	0225 n	0226 o
0160	0227 p	0230 q	0231 r	0242 s	0243 t	0244 u	0245 v	0246 w
0170	0247 x	0250 y	0251 z	0300 {	0117	0320 }	0137 ~	0007 DEL
0200	0040 DS	0041 SOS	0042 FS	0043 WUS	0044 BYP	0025 NL	0006 RNL	0027 POC
0210	0050 SA	0051 SFE	0052 SM	0053 CSP	0054 MFA	0011 SPS	0012 RPT	0033 CU1
0220	0060	0061	0032 UBS	0063 IR	0064 PP	0065 TRN	0066 NBS	0010 GE
0230	0070 SBS	0071 IT	0072 RFF	0073 CU3	0004 SEL	0024 RES	0076	0341
0240	0101	0102	0103	0104	0105	0106	0107	0110
0250	0111	0121	0122	0123	0124	0125	0126	0127
0260	0130	0131	0142	0143	0144	0145	0146	0147
0270	0150	0151	0160	0161	0162	0163	0164	0165
0300	0166	0167	0170	0200	0212	0213	0214	0215
0310	0216	0217	0220	0152	0233	0234	0235	0236
0320	0237	0240	0252	0253	0254	0112 ¢	0256	0257
0330	0260	0261	0262	0263	0264	0265	0266	0267
0340	0270	0271	0272	0273	0274	0241	0276	0277
0350	0312	0313	0314 ⠉	0315	0316 ⠄	0317	0332	0333
0360	0334	0335	0336	0337	0352	0353	0354 ⠊	0355
0370	0356	0357	0372	0373	0374	0375	0376	0377 EO

Table 4-7 ASCII to IBM EBCDIC Conversion

	0	1	2	3	4	5	6	7
0000	0000 NUL	0001 SOH	0002 STX	0003 ETX	0067 EOT	0055 ENQ	0056 ACK	0057 BEL
0010	0026 BS	0005 HT	0045 LF	0013 VT	0014 FF	0015 CR	0016 SO	0017 SI
0020	0020 DLE	0021 DC1	0022 DC2	0023 DC3	0074 DC4	0075 NAK	0062 SYN	0046 ETB
0030	0030 CAN	0031 EM	0077 SUB	0047 ESC	0034 IFS	0035 IGS	0036 IRS	0037 ITB
0040	0100 Sp	0132 !	0177 "	0173 #	0133 \$	0154 %	0120 &	0175 '
0050	0115 (0135)	0134 *	0116 +	0153 ,	0140 -	0113 .	0141 /
0060	0360 0	0361 1	0362 2	0363 3	0364 4	0365 5	0366 6	0367 7
0070	0370 8	0371 9	0172 :	0136 ;	0114 <	0176 =	0156 >	0157 ?
0100	0174 @	0301 A	0302 B	0303 C	0304 D	0305 E	0306 F	0307 G
0110	0310 H	0311 I	0321 J	0322 K	0323 L	0324 M	0325 N	0326 O
0120	0327 P	0330 Q	0331 R	0342 S	0343 T	0344 U	0345 V	0346 W
0130	0347 X	0350 Y	0351 Z	0255 [0340 \	0275]	0137 ~	0155 _
0140	0171 `	0201 a	0202 b	0203 c	0204 d	0205 e	0206 f	0207 g
0150	0210 h	0211 i	0221 j	0222 k	0223]	0224 m	0225 n	0226 o
0160	0227 p	0230 q	0231 r	0242 s	0243 t	0244 u	0245 v	0246 w
0170	0247 x	0250 y	0251 z	0300 {	0117	0320 }	0241	0007 DEL
0200	0040 DS	0041 SOS	0042 FS	0043 WUS	0044 BYP	0025 NL	0006 RNL	0027 POC
0210	0050 SA	0051 SFE	0052 SM	0053 CSP	0054 MFA	0011 SPS	0012 RPT	0033 CU1
0220	0060	0061	0032 UBS	0063 IR	0064 PP	0065 TRN	0066 NBS	0010 GE
0230	0070 SBS	0071 IT	0072 RFF	0073 CU3	0004 SEL	0024 RES	0076	0341
0240	0101	0102	0103	0104	0105	0106	0107	0110
0250	0111	0121	0122	0123	0124	0125	0126	0127
0260	0130	0131	0142	0143	0144	0145	0146	0147
0270	0150	0151	0160	0161	0162	0163	0164	0165
0300	0166	0167	0170	0200	0212	0213	0214	0215
0310	0216	0217	0220	0232	0233	0234	0235	0236
0320	0237	0240	0252	0253	0254	0255 [0256	0257
0330	0260	0261	0262	0263	0264	0265	0266	0267
0340	0270	0271	0272	0273	0274	0275]	0276	0277
0350	0312	0313	0314 ↴	0315	0316 ↵	0317	0332	0333
0360	0334	0335	0336	0337	0352	0353	0354 ↪	0355
0370	0356	0357	0372	0373	0374	0375	0376	0377 EO

11737 STDIN

11738 If no **if=** operand is specified, the standard input shall be used. See the INPUT FILES section.

11739 INPUT FILES

11740 The input file can be any file type.

11741 ENVIRONMENT VARIABLES

11742 The following environment variables shall affect the execution of *dd*:

11743 **LANG** Provide a default value for the internationalization variables that are unset or null.
11744 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
11745 Internationalization Variables for the precedence of internationalization variables
11746 used to determine the values of locale categories.)

11747 **LC_ALL** If set to a non-empty string value, override the values of all the other
11748 internationalization variables.

11749 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
11750 characters (for example, single-byte as opposed to multi-byte characters in
11751 arguments and input files), the classification of characters as uppercase or
11752 lowercase, and the mapping of characters from one case to the other.

11753 LC_MESSAGES

11754 Determine the locale that should be used to affect the format and contents of
11755 diagnostic messages written to standard error and informative messages written to
11756 standard output.

11757 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

11758 ASYNCHRONOUS EVENTS

11759 For SIGINT, the *dd* utility shall interrupt its current processing, write status information to
11760 standard error, and exit as though terminated by SIGINT. It shall take the standard action for all
11761 other signals; see the ASYNCHRONOUS EVENTS section in Section 1.11 (on page 20).

11762 STDOUT

11763 If no **of=** operand is specified, the standard output shall be used. The nature of the output
11764 depends on the operands selected.

11765 STDERR

11766 On completion, *dd* shall write the number of input and output blocks to standard error. In the
11767 POSIX locale the following formats shall be used:

11768 "%u+%u records in\n", <number of whole input blocks>,
11769 <number of partial input blocks>

11770 "%u+%u records out\n", <number of whole output blocks>,
11771 <number of partial output blocks>

11772 A partial input block is one for which *read()* returned less than the input block size. A partial
11773 output block is one that was written with fewer bytes than specified by the output block size.

11774 In addition, when there is at least one truncated block, the number of truncated blocks shall be
11775 written to standard error. In the POSIX locale, the format shall be:

11776 "%u truncated %s\n", <number of truncated blocks>, "record" (if
11777 <number of truncated blocks> is one) "records" (otherwise)

11778 Diagnostic messages may also be written to standard error.

11779 OUTPUT FILES

11780 If the **of=** operand is used, the output shall be the same as described in the STDOUT section.

11781 EXTENDED DESCRIPTION

11782 None.

11783 EXIT STATUS

11784 The following exit values shall be returned:

11785 0 The input file was copied successfully.

11786 >0 An error occurred.

11787 CONSEQUENCES OF ERRORS

11788 If an input error is detected and the **noerror** conversion has not been specified, any partial
11789 output block shall be written to the output file, a diagnostic message shall be written, and the
11790 copy operation shall be discontinued. If some other error is detected, a diagnostic message shall
11791 be written and the copy operation shall be discontinued.

11792 APPLICATION USAGE

11793 The input and output block size can be specified to take advantage of raw physical I/O.

11794 There are many different versions of the EBCDIC codesets. The ASCII and EBCDIC conversions
11795 specified for the *dd* utility perform conversions for the version specified by the tables.

11796 EXAMPLES

11797 The following command:

11798 `dd if=/dev/rmt0h of=/dev/rmt1h`

11799 copies from tape drive 0 to tape drive 1, using a common historical device naming convention.

11800 The following command:

11801 `dd ibs=10 skip=1`

11802 strips the first 10 bytes from standard input.

11803 This example reads an EBCDIC tape blocked ten 80-byte EBCDIC card images per block into the
11804 ASCII file **x**:

11805 `dd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase`

11806 RATIONALE

11807 The OPTIONS section is listed as “None” because there are no options recognized by historical
11808 *dd* utilities. Certainly, many of the operands could have been designed to use the Utility Syntax
11809 Guidelines, which would have resulted in the classic hyphenated option letters. In this version
11810 of this volume of IEEE Std 1003.1-2001, *dd* retains its curious JCL-like syntax due to the large
11811 number of applications that depend on the historical implementation.

11812 A suggested implementation technique for **conv=noerror,sync** is to zero (or <space>-fill, if
11813 **blocking** or **unblocking**) the input buffer before each read and to write the contents of the input
11814 buffer to the output even after an error. In this manner, any data transferred to the input buffer
11815 before the error was detected is preserved. Another point is that a failed read on a regular file or
11816 a disk generally does not increment the file offset, and *dd* must then seek past the block on which
11817 the error occurred; otherwise, the input error occurs repetitively. When the input is a magnetic
11818 tape, however, the tape normally has passed the block containing the error when the error is
11819 reported, and thus no seek is necessary.

11820 The default **ibs=** and **obs=** sizes are specified as 512 bytes because there are historical (largely
11821 portable) scripts that assume these values. If they were left unspecified, unusual results could

11822 occur if an implementation chose an odd block size.

11823 Historical implementations of *dd* used *creat()* when processing **of=***file*. This makes the **seek=**
11824 operand unusable except on special files. The **conv=notrunc** feature was added because more
11825 recent BSD-based implementations use *open()* (without O_TRUNC) instead of *creat()*, but they
11826 fail to delete output file contents after the data copied.

11827 The **w** multiplier (historically meaning *word*), is used in System V to mean 2 and in 4.2 BSD to
11828 mean 4. Since *word* is inherently non-portable, its use is not supported by this volume of
11829 IEEE Std 1003.1-2001.

11830 Standard EBCDIC does not have the characters '[' and ']'. The values used in the table are
11831 taken from a common print train that does contain them. Other than those characters, the print
11832 train values are not filled in, but appear to provide some of the motivation for the historical
11833 choice of translations reflected here.

11834 The Standard EBCDIC table provides a 1:1 translation for all 256 bytes.

11835 The IBM EBCDIC table does not provide such a translation. The marked cells in the tables differ
11836 in such a way that:

- 11837 1. EBCDIC 0112 ('¤') and 0152 (broken pipe) do not appear in the table.
- 11838 2. EBCDIC 0137 ('¬') translates to/from ASCII 0236 ('^'). In the standard table, EBCDIC
11839 0232 (no graphic) is used.
- 11840 3. EBCDIC 0241 ('~') translates to/from ASCII 0176 ('~'). In the standard table, EBCDIC
11841 0137 ('¬') is used.
- 11842 4. 0255 ('[') and 0275 ('])' appear twice, once in the same place as for the standard table
11843 and once in place of 0112 ('¤') and 0241 ('~').

11844 In net result:

11845 EBCDIC 0275 (']') displaced EBCDIC 0241 ('~') in cell 0345.
11846 That displaced EBCDIC 0137 ('¬') in cell 0176.
11847 That displaced EBCDIC 0232 (no graphic) in cell 0136.
11848 That replaced EBCDIC 0152 (broken pipe) in cell 0313.
11849 EBCDIC 0255 ('[') replaced EBCDIC 0112 ('¤').

11850 This translation, however, reflects historical practice that (ASCII) '~' and '¬' were often
11851 mapped to each other, as were '[' and '¤'; and ']' and (EBCDIC) '~'.

11852 The **cbs** operand is required if any of the **ascii**, **ebcdic**, or **ibm** operands are specified. For the
11853 **ascii** operand, the input is handled as described for the **unblock** operand except that characters
11854 are converted to ASCII before the trailing <space>s are deleted. For the **ebcdic** and **ibm**
11855 operands, the input is handled as described for the **block** operand except that the characters are
11856 converted to EBCDIC or IBM EBCDIC after the trailing <space>s are added.

11857 The **block** and **unblock** keywords are from historical BSD practice.

11858 The consistent use of the word **record** in standard error messages matches most historical
11859 practice. An earlier version of System V used **block**, but this has been updated in more recent
11860 releases.

11861 Early proposals only allowed two numbers separated by **x** to be used in a product when
11862 specifying **bs=**, **cbs=**, **ibs=**, and **obs=** sizes. This was changed to reflect the historical practice of
11863 allowing multiple numbers in the product as provided by Version 7 and all releases of System V

11864 and BSD.

11865 A change to the **swab** conversion is required to match historical practice and is the result of IEEE
11866 PASC Interpretations 1003.2 #03 and #04, submitted for the ISO POSIX-2:1993 standard.

11867 A change to the handling of SIGINT is required to match historical practice and is the result of IEEE
11868 PASC Interpretation 1003.2 #06 submitted for the ISO POSIX-2:1993 standard.

11869 FUTURE DIRECTIONS

11870 None.

11871 SEE ALSO

11872 Section 1.11 (on page 20), *sed*, *tr*

11873 CHANGE HISTORY

11874 First released in Issue 2.

11875 Issue 5

11876 The second paragraph of the **cbs=** description is reworded and marked EX.

11877 The FUTURE DIRECTIONS section is added.

11878 Issue 6

11879 Changes are made to **swab** conversion and SIGINT handling to align with the IEEE P1003.2b
11880 draft standard.

11881 The normative text is reworded to avoid use of the term “must” for application requirements.

11882 IEEE PASC Interpretation 1003.2 #209 is applied, clarifying the interaction between **dd of=file** and
11883 **conv=notrunc**.

11884 NAME

11885 delta — make a delta (change) to an SCCS file (**DEVELOPMENT**)

11886 SYNOPSIS

11887 XSI `delta [-nps][-g list][-m mrlist][-r SID][-y[comment]] file...`
11888

11889 DESCRIPTION

11890 The *delta* utility shall be used to permanently introduce into the named SCCS files changes that
11891 were made to the files retrieved by *get* (called the *g-files*, or generated files).

11892 OPTIONS

11893 The *delta* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
11894 12.2, Utility Syntax Guidelines, except that the **-y** option has an optional option-argument. This
11895 optional option-argument shall not be presented as a separate argument.

11896 The following options shall be supported:

- 11897 **-r SID** Uniquely identify which delta is to be made to the SCCS file. The use of this option
11898 shall be necessary only if two or more outstanding *get* commands for editing (*get*
11899 **-e**) on the same SCCS file were done by the same person (login name). The SID
11900 value specified with the **-r** option can be either the SID specified on the *get*
11901 command line or the SID to be made as reported by the *get* utility; see *get* (on page
11902 473).
- 11903 **-s** Suppress the report to standard output of the activity associated with each *file*.
11904 See the STDOUT section.
- 11905 **-n** Specify retention of the edited *g-file* (normally removed at completion of delta
11906 processing).
- 11907 **-g list** Specify a *list* (see *get* for the definition of *list*) of deltas that shall be ignored when
11908 the file is accessed at the change level (SID) created by this delta.
- 11909 **-m mrlist** Specify a modification request (MR) number that the application shall supply as
11910 the reason for creating the new delta. This shall be used if the SCCS file has the **v**
11911 flag set; see *admin*.
- 11912 If **-m** is not used and '**-**' is not specified as a file argument, and the standard
11913 input is a terminal, the prompt described in the STDOUT section shall be written
11914 to standard output before the standard input is read; if the standard input is not a
11915 terminal, no prompt shall be issued.
- 11916 MRs in a list shall be separated by <blank>s or escaped <newline>s. An
11917 unescaped <newline> shall terminate the MR list. The escape character is
11918 <backslash>.
- 11919 If the **v** flag has a value, it shall be taken to be the name of a program which
11920 validates the correctness of the MR numbers. If a non-zero exit status is returned
11921 from the MR number validation program, the *delta* utility shall terminate. (It is
11922 assumed that the MR numbers were not all valid.)
- 11923 **-y[comment]** Describe the reason for making the delta. The *comment* shall be an arbitrary group
11924 of lines that would meet the definition of a text file. Implementations shall support
11925 *comments* from zero to 512 bytes and may support longer values. A null string
11926 (specified as either **-y**, **-y" "**, or in response to a prompt for a comment) shall be
11927 considered a valid *comment*.

11928 If **-y** is not specified and '**-**' is not specified as a file argument, and the standard
11929 input is a terminal, the prompt described in the STDOUT section shall be written
11930 to standard output before the standard input is read; if the standard input is not a
11931 terminal, no prompt shall be issued. An unescaped <newline> shall terminate the
11932 comment text. The escape character is <backslash>.

11933 The **-y** option shall be required if the *file* operand is specified as '**-**'.

11934 **-p** Write (to standard output) the SCCS file differences before and after the delta is
11935 applied in *diff* format; see *diff*.

11936 OPERANDS

11937 The following operand shall be supported:

11938 *file* A pathname of an existing SCCS file or a directory. If *file* is a directory, the *delta*
11939 utility shall behave as though each file in the directory were specified as a named
11940 file, except that non-SCCS files (last component of the pathname does not begin
11941 with s.) and unreadable files shall be silently ignored.

11942 If exactly one *file* operand appears, and it is '**-**', the standard input shall be read;
11943 each line of the standard input shall be taken to be the name of an SCCS file to be
11944 processed. Non-SCCS files and unreadable files shall be silently ignored.

11945 STDIN

11946 The standard input shall be a text file used only in the following cases:

- 11947 • To read an *mrlist* or a *comment* (see the **-m** and **-y** options).
- 11948 • A *file* operand shall be specified as '**-**'. In this case, the **-y** option must be used to specify
11949 the comment, and if the SCCS file has the **v** flag set, the **-m** option must also be used to
11950 specify the MR list.

11951 INPUT FILES

11952 Input files shall be text files whose data is to be included in the SCCS files. If the first character of
11953 any line of an input file is <SOH> in the POSIX locale, the results are unspecified. If this file
11954 contains more than 99 999 lines, the number of lines recorded in the header for this file shall be
11955 99 999 for this delta.

11956 ENVIRONMENT VARIABLES

11957 The following environment variables shall affect the execution of *delta*:

11958 **LANG** Provide a default value for the internationalization variables that are unset or null.
11959 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
11960 Internationalization Variables for the precedence of internationalization variables
11961 used to determine the values of locale categories.)

11962 **LC_ALL** If set to a non-empty string value, override the values of all the other
11963 internationalization variables.

11964 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
11965 characters (for example, single-byte as opposed to multi-byte characters in
11966 arguments and input files).

11967 **LC_MESSAGES**

11968 Determine the locale that should be used to affect the format and contents of
11969 diagnostic messages written to standard error, and informative messages written
11970 to standard output.

11971 **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

11972 **TZ** Determine the timezone in which the time and date are written in the SCCS file. If
11973 the *TZ* variable is unset or NULL, an unspecified system default timezone is used.

11974 ASYNCHRONOUS EVENTS

11975 If SIGINT is caught, temporary files shall be cleaned up and *delta* shall exit with a non-zero exit
11976 code. The standard action shall be taken for all other signals; see Section 1.11 (on page 20).

11977 STDOUT

11978 The standard output shall be used only for the following messages in the POSIX locale:

- Prompts (see the **-m** and **-y** options) in the following formats:

11980 "MRs? "

11981 "comments? "

11982 The MR prompt, if written, shall always precede the comments prompt.

- A report of each file's activities (unless the **-s** option is specified) in the following format:

11984 "%s\n%d inserted\n%d deleted\n%d unchanged\n", <New SID>,

11985 <number of lines inserted>, <number of lines deleted>,

11986 <number of lines unchanged>

11987 STDERR

11988 The standard error shall be used only for diagnostic messages.

11989 OUTPUT FILES

11990 Any SCCS files updated shall be files of an unspecified format.

11991 EXTENDED DESCRIPTION

11992 System Date and Time

11993 When a delta is added to an SCCS file, the system date and time shall be recorded for the new
11994 delta. If a *get* is performed using an SCCS file with a date recorded apparently in the future, the
11995 behavior is unspecified.

11996 EXIT STATUS

11997 The following exit values shall be returned:

11998 0 Successful completion.

11999 >0 An error occurred.

12000 CONSEQUENCES OF ERRORS

12001 Default.

12002 APPLICATION USAGE

12003 Problems can arise if the system date and time have been modified (for example, put forward
12004 and then back again, or unsynchronized clocks across a network) and can also arise when
12005 different values of the *TZ* environment variable are used.

12006 Problems of a similar nature can also arise for the operation of the *get* utility, which records the
12007 date and time in the file body.

12008 EXAMPLES

12009 None.

12010 RATIONALE

12011 None.

12012 FUTURE DIRECTIONS

12013 None.

12014 SEE ALSO

12015 Section 1.11 (on page 20), *admin*, *diff*, *get*, *prs*, *rmdel*

12016 CHANGE HISTORY

12017 First released in Issue 2.

12018 Issue 5

12019 The output format description in the STDOUT section is corrected.

12020 Issue 6

12021 The APPLICATION USAGE section is added.

12022 The normative text is reworded to avoid use of the term “must” for application requirements.

12023 The Open Group Base Resolution bwg2001-007 is applied as follows:

12024 • The use of ‘–’ as a file argument is clarified.

12025 • The use of STDIN is added.

12026 • The ASYNCHRONOUS EVENTS section is updated to remove the implicit requirement that
12027 implementations re-signal themselves when catching a normally fatal signal.

12028 • New text is added to the INPUT FILES section warning that the maximum lines recorded in
12029 the file is 99 999.

12030 New text is added to the EXTENDED DESCRIPTION and APPLICATION USAGE sections
12031 regarding how the system date and time may be taken into account, and the TZ environment
12032 variable is added to the ENVIRONMENT VARIABLES section as per The Open Group Base
12033 Resolution bwg2001-007.

12034 NAME

12035 df — report free disk space

12036 SYNOPSIS

12037 UP XSI df [-k] [-P | -t] [file...]

12038

12039 DESCRIPTION

12040 XSI The *df* utility shall write the amount of available space and file slots for file systems on which the invoking user has appropriate read access. File systems shall be specified by the *file* operands; when none are specified, information shall be written for all file systems. The format of the default output from *df* is unspecified, but all space figures are reported in 512-byte units, unless the **-k** option is specified. This output shall contain at least the file system names, amount of available space on each of these file systems, and the number of free file slots, or *inodes*, available; when **-t** is specified, the output shall contain the total allocated space as well.

12047 OPTIONS

12048 The *df* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

12050 The following options shall be supported:

12051 **-k** Use 1024-byte units, instead of the default 512-byte units, when writing space figures.

12053 **-P** Produce output in the format described in the STDOUT section.

12054 XSI **-t** Include total allocated-space figures in the output.

12055 OPERANDS

12056 The following operand shall be supported:

12057 *file* A pathname of a file within the hierarchy of the desired file system. If a file other than a FIFO, a regular file, a directory, or a special file representing the device containing the file system (for example, */dev/dsk/0s1*) is specified, the results are unspecified. Otherwise, *df* shall write the amount of free space in the file system containing the specified *file* operand.

12062 STDIN

12063 Not used.

12064 INPUT FILES

12065 None.

12066 ENVIRONMENT VARIABLES

12067 The following environment variables shall affect the execution of *df*:

12068 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

12072 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

12074 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

- 12077 ***LC_MESSAGES***
 12078 Determine the locale that should be used to affect the format and contents of
 12079 diagnostic messages written to standard error and informative messages written to
 12080 standard output.
- 12081 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.
- 12082 **ASYNCHRONOUS EVENTS**
- 12083 Default.
- 12084 **STDOUT**
- 12085 When both the **-k** and **-P** options are specified, the following header line shall be written (in the
 12086 POSIX locale):
 "Filesystem 1024-blocks Used Available Capacity Mounted on\n"
- 12088 When the **-P** option is specified without the **-k** option, the following header line shall be written
 12089 (in the POSIX locale):
 "Filesystem 512-blocks Used Available Capacity Mounted on\n"
- 12091 The implementation may adjust the spacing of the header line and the individual data lines so
 12092 that the information is presented in orderly columns.
- 12093 The remaining output with **-P** shall consist of one line of information for each specified file
 12094 system. These lines shall be formatted as follows:
- 12095 "%s %d %d %d%% %s\n", <file system name>, <total space>,
 12096 <space used>, <space free>, <percentage used>,
 12097 <file system root>
- 12098 In the following list, all quantities expressed in 512-byte units (1 024-byte when **-k** is specified)
 12099 shall be rounded up to the next higher unit. The fields are:
- 12100 **<file system name>**
 12101 The name of the file system, in an implementation-defined format.
- 12102 **<total space>** The total size of the file system in 512-byte units. The exact meaning of this figure
 12103 is implementation-defined, but should include **<space used>**, **<space free>**, plus any
 12104 space reserved by the system not normally available to a user.
- 12105 **<space used>** The total amount of space allocated to existing files in the file system, in 512-byte
 12106 units.
- 12107 **<space free>** The total amount of space available within the file system for the creation of new
 12108 files by unprivileged users, in 512-byte units. When this figure is less than or equal
 12109 to zero, it shall not be possible to create any new files on the file system without
 12110 first deleting others, unless the process has appropriate privileges. The figure
 12111 written may be less than zero.
- 12112 **<percentage used>**
 12113 The percentage of the normally available space that is currently allocated to all
 12114 files on the file system. This shall be calculated using the fraction:
 12115 <space used>/(<space used>+<space free>)
- 12116 expressed as a percentage. This percentage may be greater than 100 if **<space free>**
 12117 is less than zero. The percentage value shall be expressed as a positive integer,
 12118 with any fractional result causing it to be rounded to the next highest integer.

12119 <file system root>
12120 The directory below which the file system hierarchy appears.

12121 XSI The output format is unspecified when **-t** is used.

12122 **STDERR**

12123 The standard error shall be used only for diagnostic messages.

12124 **OUTPUT FILES**

12125 None.

12126 **EXTENDED DESCRIPTION**

12127 None.

12128 **EXIT STATUS**

12129 The following exit values shall be returned:

12130 0 Successful completion.

12131 >0 An error occurred.

12132 **CONSEQUENCES OF ERRORS**

12133 Default.

12134 **APPLICATION USAGE**

12135 On most systems, the “name of the file system, in an implementation-defined format” is the
12136 special file on which the file system is mounted.

12137 On large file systems, the calculation specified for percentage used can create huge rounding
12138 errors.

12139 **EXAMPLES**

12140 1. The following example writes portable information about the **/usr** file system:

12141 df -P /usr

12142 2. Assuming that **/usr/src** is part of the **/usr** file system, the following produces the same
12143 output as the previous example:

12144 df -P /usr/src

12145 **RATIONALE**

12146 The behavior of *df* with the **-P** option is the default action of the 4.2 BSD *df* utility. The uppercase
12147 **-P** was selected to avoid collision with a known industry extension using **-p**.

12148 Historical *df* implementations vary considerably in their default output. It was therefore
12149 necessary to describe the default output in a loose manner to accommodate all known historical
12150 implementations and to add a portable option (**-P**) to provide information in a portable format.

12151 The use of 512-byte units is historical practice and maintains compatibility with *ls* and other
12152 utilities in this volume of IEEE Std 1003.1-2001. This does not mandate that the file system itself
12153 be based on 512-byte blocks. The **-k** option was added as a compromise measure. It was agreed
12154 by the standard developers that 512 bytes was the best default unit because of its complete
12155 historical consistency on System V (versus the mixed 512/1 024-byte usage on BSD systems), and
12156 that a **-k** option to switch to 1 024-byte units was a good compromise. Users who prefer the
12157 more logical 1 024-byte quantity can easily alias *df* to *df -k* without breaking many historical
12158 scripts relying on the 512-byte units.

12159 It was suggested that *df* and the various related utilities be modified to access a **BLOCKSIZE**
12160 environment variable to achieve consistency and user acceptance. Since this is not historical
12161 practice on any system, it is left as a possible area for system extensions and will be re-evaluated

12162 in a future version if it is widely implemented.

12163 FUTURE DIRECTIONS

12164 None.

12165 SEE ALSO

12166 *find*

12167 CHANGE HISTORY

12168 First released in Issue 2.

12169 Issue 6

12170 This utility is marked as part of the User Portability Utilities option.

12171 NAME

12172 **diff** — compare two files

12173 SYNOPSIS

12174 **diff** [**-c** | **-e** | **-f** | **-C** *n*] [**-br**] *file1* *file2*

12175 DESCRIPTION

12176 The *diff* utility shall compare the contents of *file1* and *file2* and write to standard output a list of
12177 changes necessary to convert *file1* into *file2*. This list should be minimal. No output shall be
12178 produced if the files are identical.

12179 OPTIONS

12180 The *diff* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
12181 12.2, Utility Syntax Guidelines.

12182 The following options shall be supported:

12183 **-b** Cause any amount of white space at the end of a line to be treated as a single
12184 <newline> (that is, the white-space characters preceding the <newline> are
12185 ignored) and other strings of white-space characters, not including <newline>s, to
12186 compare equal.

12187 **-c** Produce output in a form that provides three lines of context.

12188 **-C** *n* Produce output in a form that provides *n* lines of context (where *n* shall be
12189 interpreted as a positive decimal integer).

12190 **-e** Produce output in a form suitable as input for the *ed* utility, which can then be
12191 used to convert *file1* into *file2*.

12192 **-f** Produce output in an alternative form, similar in format to **-e**, but not intended to
12193 be suitable as input for the *ed* utility, and in the opposite order.

12194 **-r** Apply *diff* recursively to files and directories of the same name when *file1* and *file2*
12195 are both directories.

12196 OPERANDS

12197 The following operands shall be supported:

12198 *file1*, *file2* A pathname of a file to be compared. If either the *file1* or *file2* operand is '**-**', the
12199 standard input shall be used in its place.

12200 If both *file1* and *file2* are directories, *diff* shall not compare block special files, character special
12201 files, or FIFO special files to any files and shall not compare regular files to directories. Further
12202 details are as specified in **Diff Directory Comparison Format** (on page 318). The behavior of *diff*
12203 on other file types is implementation-defined when found in directories.

12204 If only one of *file1* and *file2* is a directory, *diff* shall be applied to the non-directory file and the file
12205 contained in the directory file with a filename that is the same as the last component of the non-
12206 directory file.

12207 STDIN

12208 The standard input shall be used only if one of the *file1* or *file2* operands references standard
12209 input. See the INPUT FILES section.

12210 INPUT FILES

12211 The input files may be of any type.

12212 ENVIRONMENT VARIABLES

12213 The following environment variables shall affect the execution of *diff*:

12214	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
12218	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
12220	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).
12223	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.
12227	<i>LC_TIME</i>	Determine the locale for affecting the format of file timestamps written with the -C and -c options.
12229 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
12230	<i>TZ</i>	Determine the timezone used for calculating file timestamps written with the -C and -c options. If <i>TZ</i> is unset or null, an unspecified default timezone shall be used.

12233 ASYNCHRONOUS EVENTS

12234 Default.

12235 STDOUT

12236 Diff Directory Comparison Format

12237 If both *file1* and *file2* are directories, the following output formats shall be used.

12238 In the POSIX locale, each file that is present in only one directory shall be reported using the following format:

12240 "Only in %s: %s\n", <directory pathname>, <filename>

12241 In the POSIX locale, subdirectories that are common to the two directories may be reported with the following format:

12243 "Common subdirectories: %s and %s\n", <directory1 pathname>,
12244 <directory2 pathname>

12245 For each file common to the two directories if the two files are not to be compared, the following
12246 format shall be used in the POSIX locale:

12247 "File %s is a %s while file %s is a %s\n", <directory1 pathname>,
12248 <file type of directory1 pathname>, <directory2 pathname>,
12249 <file type of directory2 pathname>

12250 For each file common to the two directories, if the files are compared and are identical, no output
12251 shall be written. If the two files differ, the following format is written:

12252 "diff %s %s %s\n", <diff_options>, <filename1>, <filename2>

12253 where <diff_options> are the options as specified on the command line.
12254 All directory pathnames listed in this section shall be relative to the original command line
12255 arguments. All other names of files listed in this section shall be filenames (pathname
12256 components).

12257 Diff Binary Output Format

12258 In the POSIX locale, if one or both of the files being compared are not text files, an unspecified
12259 format shall be used that contains the pathnames of two files being compared and the string
12260 "differ".

12261 If both files being compared are text files, depending on the options specified, one of the
12262 following formats shall be used to write the differences.

12263 Diff Default Output Format

12264 The default (without -e, -f, -c, or -C options) *diff* utility output shall contain lines of these
12265 forms:

12266 "%da%d\n", <num1>, <num2>
12267 "%da%d,%d\n", <num1>, <num2>, <num3>
12268 "%dd%d\n", <num1>, <num2>
12269 "%d,%dd%d\n", <num1>, <num2>, <num3>
12270 "%dc%d\n", <num1>, <num2>
12271 "%d,%dc%d\n", <num1>, <num2>, <num3>
12272 "%dc%d,%d\n", <num1>, <num2>, <num3>
12273 "%d,%dc%d,%d\n", <num1>, <num2>, <num3>, <num4>

12274 These lines resemble *ed* subcommands to convert *file1* into *file2*. The line numbers before the
12275 action letters shall pertain to *file1*; those after shall pertain to *file2*. Thus, by exchanging a for d
12276 and reading the line in reverse order, one can also determine how to convert *file2* into *file1*. As in
12277 *ed*, identical pairs (where num1= num2) are abbreviated as a single number.

12278 Following each of these lines, *diff* shall write to standard output all lines affected in the first file
12279 using the format:

12280 "<Δ%>s", <line>

12281 and all lines affected in the second file using the format:

12282 ">Δ%>s", <line>

12283 If there are lines affected in both *file1* and *file2* (as with the c subcommand), the changes are
12284 separated with a line consisting of three hyphens:

12285 "---\n"

12286 **Diff -e Output Format**

12287 With the **-e** option, a script shall be produced that shall, when provided as input to *ed*, along
12288 with an appended **w** (write) command, convert *file1* into *file2*. Only the **a** (append), **c** (change), **d**
12289 (delete), **i** (insert), and **s** (substitute) commands of *ed* shall be used in this script. Text lines,
12290 except those consisting of the single character period ('.'), shall be output as they appear in the
12291 file.

12292 **Diff -f Output Format**

12293 With the **-f** option, an alternative format of script shall be produced. It is similar to that
12294 produced by **-e**, with the following differences:

- 12295 1. It is expressed in reverse sequence; the output of **-e** orders changes from the end of the file
12296 to the beginning; the **-f** from beginning to end.
- 12297 2. The command form *<lines> <command-letter>* used by **-e** is reversed. For example,
12298 *10c* with **-e** would be *c10* with **-f**.
- 12299 3. The form used for ranges of line numbers is *<space>*-separated, rather than comma-
12300 separated.

12301 **Diff -c or -C Output Format**

12302 With the **-c** or **-C** option, the output format shall consist of affected lines along with
12303 surrounding lines of context. The affected lines shall show which ones need to be deleted or
12304 changed in *file1*, and those added from *file2*. With the **-c** option, three lines of context, if
12305 available, shall be written before and after the affected lines. With the **-C** option, the user can
12306 specify how many lines of context are written. The exact format follows.

12307 The name and last modification time of each file shall be output in the following format:

```
12308       "*** %s %s\n", file1, <file1 timestamp>
12309       "--- %s %s\n", file2, <file2 timestamp>
```

12310 Each *<file>* field shall be the pathname of the corresponding file being compared. The pathname
12311 written for standard input is unspecified.

12312 In the POSIX locale, each *<timestamp>* field shall be equivalent to the output from the following
12313 command:

```
12314       date "+%a %b %e %T %Y"
```

12315 without the trailing *<newline>*, executed at the time of last modification of the corresponding
12316 file (or the current time, if the file is standard input).

12317 Then, the following output formats shall be applied for every set of changes.

12318 First, a line shall be written in the following format:

```
12319       "*****\n"
```

12320 Next, the range of lines in *file1* shall be written in the following format:

```
12321       "*** %d,%d ****\n", <beginning line number>, <ending line number>
```

12322 Next, the affected lines along with lines of context (unaffected lines) shall be written. Unaffected
12323 lines shall be written in the following format:

```
12324       "ΔΔ%s" , <unaffected_line>
```

12325 Deleted lines shall be written as:
12326 " -Δ%s " , <deleted_line>
12327 Changed lines shall be written as:
12328 " !Δ%s " , <changed_line>
12329 Next, the range of lines in *file2* shall be written in the following format:
12330 " --- %d,%d ----\n" , <beginning line number> , <ending line number>
12331 Then, lines of context and changed lines shall be written as described in the previous formats.
12332 Lines added from *file2* shall be written in the following format:
12333 " +Δ%s " , <added_line>

12334 **STDERR**
12335 The standard error shall be used only for diagnostic messages.

12336 **OUTPUT FILES**
12337 None.

12338 **EXTENDED DESCRIPTION**
12339 None.

12340 **EXIT STATUS**
12341 The following exit values shall be returned:
12342 0 No differences were found.
12343 1 Differences were found.
12344 >1 An error occurred.

12345 **CONSEQUENCES OF ERRORS**
12346 Default.

12347 **APPLICATION USAGE**
12348 If lines at the end of a file are changed and other lines are added, *diff* output may show this as a
12349 delete and add, as a change, or as a change and add; *diff* is not expected to know which
12350 happened and users should not care about the difference in output as long as it clearly shows the
12351 differences between the files.

12352 **EXAMPLES**
12353 If **dir1** is a directory containing a directory named **x**, **dir2** is a directory containing a directory
12354 named **x**, **dir1/x** and **dir2/x** both contain files named **date.out**, and **dir2/x** contains a file named **y**,
12355 the command:
12356 *diff -r dir1 dir2*
12357 could produce output similar to:
12358 Common subdirectories: **dir1/x** and **dir2/x**
12359 Only in **dir2/x**: **y**
12360 *diff -r dir1/x/date.out dir2/x/date.out*
12361 **1cl**
12362 < Mon Jul 2 13:12:16 PDT 1990
12363 ---
12364 > Tue Jun 19 21:41:39 PDT 1990

12365 RATIONALE

12366 The **-h** option was omitted because it was insufficiently specified and does not add to
12367 applications portability.

12368 Historical implementations employ algorithms that do not always produce a minimum list of
12369 differences; the current language about making every effort is the best this volume of
12370 IEEE Std 1003.1-2001 can do, as there is no metric that could be employed to judge the quality of
12371 implementations against any and all file contents. The statement “This list should be minimal”
12372 clearly implies that implementations are not expected to provide the following output when
12373 comparing two 100-line files that differ in only one character on a single line:

```
12374 1,100c1,100
12375 all 100 lines from file1 preceded with "< "
12376 ---
12377 all 100 lines from file2 preceded with "> "
```

12378 The “Only in” messages required when the **-r** option is specified are not used by most historical
12379 implementations if the **-e** option is also specified. It is required here because it provides useful
12380 information that must be provided to update a target directory hierarchy to match a source
12381 hierarchy. The “Common subdirectories” messages are written by System V and 4.3 BSD when
12382 the **-r** option is specified. They are allowed here but are not required because they are reporting
12383 on something that is the same, not reporting a difference, and are not needed to update a target
12384 hierarchy.

12385 The **-c** option, which writes output in a format using lines of context, has been included. The
12386 format is useful for a variety of reasons, among them being much improved readability and the
12387 ability to understand difference changes when the target file has line numbers that differ from
12388 another similar, but slightly different, copy. The *patch* utility is most valuable when working
12389 with difference listings using the context format. The BSD version of **-c** takes an optional
12390 argument specifying the amount of context. Rather than overloading **-c** and breaking the Utility
12391 Syntax Guidelines for *diff*, the standard developers decided to add a separate option for
12392 specifying a context diff with a specified amount of context (**-C**). Also, the format for context
12393 *diffs* was extended slightly in 4.3 BSD to allow multiple changes that are within context lines
12394 from each other to be merged together. The output format contains an additional four asterisks
12395 after the range of affected lines in the first filename. This was to provide a flag for old programs
12396 (like old versions of *patch*) that only understand the old context format. The version of context
12397 described here does not require that multiple changes within context lines be merged, but it does
12398 not prohibit it either. The extension is upwards-compatible, so any vendors that wish to retain
12399 the old version of *diff* can do so by adding the extra four asterisks (that is, utilities that currently
12400 use *diff* and understand the new merged format will also understand the old unmerged format,
12401 but not *vice versa*).

12402 The substitute command was added as an additional format for the **-e** option. This was added to
12403 provide implementations with a way to fix the classic “dot alone on a line” bug present in many
12404 versions of *diff*. Since many implementations have fixed this bug, the standard developers
12405 decided not to standardize broken behavior, but rather to provide the necessary tool for fixing
12406 the bug. One way to fix this bug is to output two periods whenever a lone period is needed, then
12407 terminate the append command with a period, and then use the substitute command to convert
12408 the two periods into one period.

12409 The BSD-derived **-r** option was added to provide a mechanism for using *diff* to compare two file
12410 system trees. This behavior is useful, is standard practice on all BSD-derived systems, and is not
12411 easily reproducible with the *find* utility.

12412 The requirement that *diff* not compare files in some circumstances, even though they have the
12413 same name, is based on the actual output of historical implementations. The message specified

here is already in use when a directory is being compared to a non-directory. It is extended here to preclude the problems arising from running into FIFOs and other files that would cause *diff* to hang waiting for input with no indication to the user that *diff* was hung. In most common usage, *diff -r* should indicate differences in the file hierarchies, not the difference of contents of devices pointed to by the hierarchies.

Many early implementations of *diff* require seekable files. Since the System Interfaces volume of IEEE Std 1003.1-2001 supports named pipes, the standard developers decided that such a restriction was unreasonable. Note also that the allowed filename – almost always refers to a pipe.

No directory search order is specified for *diff*. The historical ordering is, in fact, not optimal, in that it prints out all of the differences at the current level, including the statements about all common subdirectories before recursing into those subdirectories.

The message:

"*diff %s %s %s\n*", <*diff_options*>, <*filename1*>, <*filename2*>

does not vary by locale because it is the representation of a command, not an English sentence.

12429 FUTURE DIRECTIONS

12430 None.

12431 SEE ALSO

12432 *cmp, comm, ed, find*

12433 CHANGE HISTORY

12434 First released in Issue 2.

12435 Issue 5

12436 The FUTURE DIRECTIONS section is added.

12437 Issue 6

12438 The following new requirements on POSIX implementations derive from alignment with the
12439 Single UNIX Specification:

- 12440 • The *-f* option is added.

12441 The output format for *-c* or *-C* format is changed to align with changes to the IEEE P1003.2b
12442 draft standard resulting from IEEE PASC Interpretation 1003.2 #71.

12443 The normative text is reworded to avoid use of the term “must” for application requirements.

12444 NAME

12445 dirname — return the directory portion of a pathname

12446 SYNOPSIS

12447 dirname *string*

12448 DESCRIPTION

12449 The *string* operand shall be treated as a pathname, as defined in the Base Definitions volume of
12450 IEEE Std 1003.1-2001, Section 3.266, Pathname. The string *string* shall be converted to the name
12451 of the directory containing the filename corresponding to the last pathname component in
12452 *string*, performing actions equivalent to the following steps in order:

- 12453 1. If *string* is //, skip steps 2 to 5.
- 12454 2. If *string* consists entirely of slash characters, *string* shall be set to a single slash character. In
12455 this case, skip steps 3 to 8.
- 12456 3. If there are any trailing slash characters in *string*, they shall be removed.
- 12457 4. If there are no slash characters remaining in *string*, *string* shall be set to a single period
12458 character. In this case, skip steps 5 to 8.
- 12459 5. If there are any trailing non-slash characters in *string*, they shall be removed.
- 12460 6. If the remaining *string* is //, it is implementation-defined whether steps 7 and 8 are skipped
12461 or processed.
- 12462 7. If there are any trailing slash characters in *string*, they shall be removed.
- 12463 8. If the remaining *string* is empty, *string* shall be set to a single slash character.

12464 The resulting string shall be written to standard output.

12465 OPTIONS

12466 None.

12467 OPERANDS

12468 The following operand shall be supported:

12469 *string* A string.

12470 STDIN

12471 Not used.

12472 INPUT FILES

12473 None.

12474 ENVIRONMENT VARIABLES

12475 The following environment variables shall affect the execution of *dirname*:

12476 *LANG* Provide a default value for the internationalization variables that are unset or null.
12477 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
12478 Internationalization Variables for the precedence of internationalization variables
12479 used to determine the values of locale categories.)

12480 *LC_ALL* If set to a non-empty string value, override the values of all the other
12481 internationalization variables.

12482 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
12483 characters (for example, single-byte as opposed to multi-byte characters in
12484 arguments).

12485	LC_MESSAGES	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.																						
12486																								
12487																								
12488 XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .																						
12489	ASYNCHRONOUS EVENTS																							
12490		Default.																						
12491	STDOUT																							
12492		The <i>dirname</i> utility shall write a line to the standard output in the following format:																						
12493		"%s\n", <resulting string>																						
12494	STDERR																							
12495		The standard error shall be used only for diagnostic messages.																						
12496	OUTPUT FILES																							
12497		None.																						
12498	EXTENDED DESCRIPTION																							
12499		None.																						
12500	EXIT STATUS																							
12501		The following exit values shall be returned:																						
12502		0 Successful completion.																						
12503		>0 An error occurred.																						
12504	CONSEQUENCES OF ERRORS																							
12505		Default.																						
12506	APPLICATION USAGE																							
12507		The definition of <i>pathname</i> specifies implementation-defined behavior for pathnames starting with two slash characters. Therefore, applications shall not arbitrarily add slashes to the beginning of a pathname unless they can ensure that there are more or less than two or are prepared to deal with the implementation-defined consequences.																						
12508																								
12509																								
12510																								
12511	EXAMPLES																							
12512	<table border="1"> <thead> <tr> <th>Command</th> <th>Results</th> </tr> </thead> <tbody> <tr> <td><i>dirname</i> /</td> <td>/</td> </tr> <tr> <td><i>dirname</i> //</td> <td>/ or //</td> </tr> <tr> <td><i>dirname</i> /a/b/</td> <td>/a</td> </tr> <tr> <td><i>dirname</i> //a//b//</td> <td>//a</td> </tr> <tr> <td><i>dirname</i></td> <td>Unspecified</td> </tr> <tr> <td><i>dirname</i> a</td> <td>. (\$? = 0)</td> </tr> <tr> <td><i>dirname</i> ""</td> <td>. (\$? = 0)</td> </tr> <tr> <td><i>dirname</i> /a</td> <td>/</td> </tr> <tr> <td><i>dirname</i> /a/b</td> <td>/a</td> </tr> <tr> <td><i>dirname</i> a/b</td> <td>a</td> </tr> </tbody> </table>	Command	Results	<i>dirname</i> /	/	<i>dirname</i> //	/ or //	<i>dirname</i> /a/b/	/a	<i>dirname</i> //a//b//	//a	<i>dirname</i>	Unspecified	<i>dirname</i> a	. (\$? = 0)	<i>dirname</i> ""	. (\$? = 0)	<i>dirname</i> /a	/	<i>dirname</i> /a/b	/a	<i>dirname</i> a/b	a	
Command	Results																							
<i>dirname</i> /	/																							
<i>dirname</i> //	/ or //																							
<i>dirname</i> /a/b/	/a																							
<i>dirname</i> //a//b//	//a																							
<i>dirname</i>	Unspecified																							
<i>dirname</i> a	. (\$? = 0)																							
<i>dirname</i> ""	. (\$? = 0)																							
<i>dirname</i> /a	/																							
<i>dirname</i> /a/b	/a																							
<i>dirname</i> a/b	a																							
12513																								
12514																								
12515																								
12516																								
12517																								
12518																								
12519																								
12520																								
12521																								
12522																								
12523	RATIONALE																							
12524		The <i>dirname</i> utility originated in System III. It has evolved through the System V releases to a version that matches the requirements specified in this description in System V Release 3. 4.3																						
12525		BSD and earlier versions did not include <i>dirname</i> .																						
12526																								
12527		The behaviors of <i>basename</i> and <i>dirname</i> in this volume of IEEE Std 1003.1-2001 have been coordinated so that when <i>string</i> is a valid pathname:																						
12528																								

12529 \$(basename "string")
12530 would be a valid filename for the file in the directory:
12531 \$(dirname "string")
12532 This would not work for the versions of these utilities in early proposals due to the way
12533 processing of trailing slashes was specified. Consideration was given to leaving processing
12534 unspecified if there were trailing slashes, but this cannot be done; the Base Definitions volume of
12535 IEEE Std 1003.1-2001, Section 3.266, Pathname allows trailing slashes. The *basename* and *dirname*
12536 utilities have to specify consistent handling for all valid pathnames.

12537 FUTURE DIRECTIONS

12538 None.

12539 SEE ALSO

12540 *basename*, Section 2.5 (on page 33)

12541 CHANGE HISTORY

12542 First released in Issue 2.

12543 NAME

12544 du — estimate file space usage

12545 SYNOPSIS

12546 UP du [-a | -s][-kx][-H | -L][file ...]

12547

12548 DESCRIPTION

12549 By default, the *du* utility shall write to standard output the size of the file space allocated to, and
12550 the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the
12551 specified files. By default, when a symbolic link is encountered on the command line or in the
12552 file hierarchy, *du* shall count the size of the symbolic link (rather than the file referenced by the
12553 link), and shall not follow the link to another portion of the file hierarchy. The size of the file
12554 space allocated to a file of type directory shall be defined as the sum total of space allocated to
12555 all files in the file hierarchy rooted in the directory plus the space allocated to the directory itself.

12556 When *du* cannot *stat()* files or *stat()* or read directories, it shall report an error condition and the
12557 final exit status is affected. Files with multiple links shall be counted and written for only one
12558 entry. The directory entry that is selected in the report is unspecified. By default, file sizes shall
12559 be written in 512-byte units, rounded up to the next 512-byte unit.

12560 OPTIONS

12561 The *du* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
12562 Utility Syntax Guidelines.

12563 The following options shall be supported:

12564 **-a** In addition to the default output, report the size of each file not of type directory in
12565 the file hierarchy rooted in the specified file. Regardless of the presence of the **-a**
12566 option, non-directories given as *file* operands shall always be listed.

12567 **-H** If a symbolic link is specified on the command line, *du* shall count the size of the
12568 file or file hierarchy referenced by the link.

12569 **-k** Write the files sizes in units of 1 024 bytes, rather than the default 512-byte units.

12570 **-L** If a symbolic link is specified on the command line or encountered during the
12571 traversal of a file hierarchy, *du* shall count the size of the file or file hierarchy
12572 referenced by the link.

12573 **-s** Instead of the default output, report only the total sum for each of the specified
12574 files.

12575 **-x** When evaluating file sizes, evaluate only those files that have the same device as
12576 the file specified by the *file* operand.

12577 Specifying more than one of the mutually-exclusive options **-H** and **-L** shall not be considered
12578 an error. The last option specified shall determine the behavior of the utility.

12579 OPERANDS

12580 The following operand shall be supported:

12581 *file* The pathname of a file whose size is to be written. If no *file* is specified, the current
12582 directory shall be used.

12583 STDIN

12584 Not used.

12585 INPUT FILES

12586 None.

12587 ENVIRONMENT VARIABLES

12588 The following environment variables shall affect the execution of *du*:

12589 **LANG** Provide a default value for the internationalization variables that are unset or null.
12590 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
12591 Internationalization Variables for the precedence of internationalization variables
12592 used to determine the values of locale categories.)

12593 **LC_ALL** If set to a non-empty string value, override the values of all the other
12594 internationalization variables.

12595 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
12596 characters (for example, single-byte as opposed to multi-byte characters in
12597 arguments).

12598 *LC_MESSAGES*

12599 Determine the locale that should be used to affect the format and contents of
12600 diagnostic messages written to standard error.

12601 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

12602 ASYNCHRONOUS EVENTS

12603 Default.

12604 STDOUT

12605 The output from *du* shall consist of the amount of space allocated to a file and the name of the
12606 file, in the following format:

12607 "%d %s\n", <size>, <pathname>

12608 STDERR

12609 The standard error shall be used only for diagnostic messages.

12610 OUTPUT FILES

12611 None.

12612 EXTENDED DESCRIPTION

12613 None.

12614 EXIT STATUS

12615 The following exit values shall be returned:

12616 0 Successful completion.

12617 >0 An error occurred.

12618 CONSEQUENCES OF ERRORS

12619 Default.

12620 APPLICATION USAGE

12621 None.

12622 EXAMPLES

12623 None.

12624 RATIONALE

12625 The use of 512-byte units is historical practice and maintains compatibility with *ls* and other
12626 utilities in this volume of IEEE Std 1003.1-2001. This does not mandate that the file system itself
12627 be based on 512-byte blocks. The **-k** option was added as a compromise measure. It was agreed
12628 by the standard developers that 512 bytes was the best default unit because of its complete
12629 historical consistency on System V (*versus* the mixed 512/1 024-byte usage on BSD systems), and
12630 that a **-k** option to switch to 1 024-byte units was a good compromise. Users who prefer the
12631 1 024-byte quantity can easily alias *du* to *du -k* without breaking the many historical scripts
12632 relying on the 512-byte units.

12633 The **-b** option was added to an early proposal to provide a resolution to the situation where
12634 System V and BSD systems give figures for file sizes in *blocks*, which is an implementation-
12635 defined concept. (In common usage, the block size is 512 bytes for System V and 1 024 bytes for
12636 BSD systems.) However, **-b** was later deleted, since the default was eventually decided as 512-
12637 byte units.

12638 Historical file systems provided no way to obtain exact figures for the space allocation given to
12639 files. There are two known areas of inaccuracies in historical file systems: cases of *indirect blocks*
12640 being used by the file system or *sparse* files yielding incorrectly high values. An indirect block is
12641 space used by the file system in the storage of the file, but that need not be counted in the space
12642 allocated to the file. A *sparse* file is one in which an *Iseek()* call has been made to a position
12643 beyond the end of the file and data has subsequently been written at that point. A file system
12644 need not allocate all the intervening zero-filled blocks to such a file. It is up to the
12645 implementation to define exactly how accurate its methods are.

12646 The **-a** and **-s** options were mutually-exclusive in the original version of *du*. The POSIX Shell
12647 and Utilities description is implied by the language in the SVID where **-s** is described as causing
12648 “only the grand total” to be reported. Some systems may produce output for **-sa**, but a Strictly
12649 Conforming POSIX Shell and Utilities Application cannot use that combination.

12650 The **-a** and **-s** options were adopted from the SVID except that the System V behavior of not
12651 listing non-directories explicitly given as operands, unless the **-a** option is specified, was
12652 considered a bug; the BSD-based behavior (report for all operands) is mandated. The default
12653 behavior of *du* in the SVID with regard to reporting the failure to read files (it produces no
12654 messages) was considered counter-intuitive, and thus it was specified that the POSIX Shell and
12655 Utilities default behavior shall be to produce such messages. These messages can be turned off
12656 with shell redirection to achieve the System V behavior.

12657 The **-x** option is historical practice on recent BSD systems. It has been adopted by this volume of
12658 IEEE Std 1003.1-2001 because there was no other historical method of limiting the *du* search to a
12659 single file hierarchy. This limitation of the search is necessary to make it possible to obtain file
12660 space usage information about a file system on which other file systems are mounted, without
12661 having to resort to a lengthy *find* and *awk* script.

12662 FUTURE DIRECTIONS

12663 None.

12664 SEE ALSO

12665 *ls*, the System Interfaces volume of IEEE Std 1003.1-2001, *stat()*

12666 CHANGE HISTORY

12667 First released in Issue 2.

12668 Issue 6

12669 This utility is marked as part of the User Portability Utilities option.

12670 The APPLICATION USAGE section is added.

12671 The obsolescent **-r** option has been removed.

12672 The Open Group Corrigendum U025/3 is applied. The *du* utility is reinstated, as it had
12673 incorrectly been marked LEGACY in Issue 5.

12674 The **-H** and **-L** options for symbolic links are added as described in the IEEE P1003.2b draft
12675 standard.

12676 NAME

12677 echo — write arguments to standard output

12678 SYNOPSIS

12679 echo [*string* ...]

12680 DESCRIPTION

12681 The *echo* utility writes its arguments to standard output, followed by a <newline>. If there are
12682 no arguments, only the <newline> is written.

12683 OPTIONS

12684 The *echo* utility shall not recognize the "—" argument in the manner specified by Guideline 10
12685 of the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines;
12686 "—" shall be recognized as a string operand.

12687 Implementations shall not support any options.

12688 OPERANDS

12689 The following operands shall be supported:

12690 *string* A string to be written to standard output. If any operand is **-n**, it shall be treated as
12691 a string, not an option. The following character sequences shall be recognized
12692 within any of the arguments:

12693 \a Write an <alert>.

12694 \b Write a <backspace>.

12695 \c Suppress the <newline> that otherwise follows the final argument in the
12696 output. All characters following the '\c' in the arguments shall be
12697 ignored.

12698 \f Write a <form-feed>.

12699 \n Write a <newline>.

12700 \r Write a <carriage-return>.

12701 \t Write a <tab>.

12702 \v Write a <vertical-tab>.

12703 \\ Write a backslash character.

12704 \0*num* Write an 8-bit value that is the zero, one, two, or three-digit octal number
12705 *num*.

12706 STDIN

12707 Not used.

12708 INPUT FILES

12709 None.

12710 ENVIRONMENT VARIABLES

12711 The following environment variables shall affect the execution of *echo*:

12712 *LANG* Provide a default value for the internationalization variables that are unset or null.
12713 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
12714 Internationalization Variables for the precedence of internationalization variables
12715 used to determine the values of locale categories.)

12716 *LC_ALL* If set to a non-empty string value, override the values of all the other
12717 internationalization variables.

12718 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
12719 characters (for example, single-byte as opposed to multi-byte characters in
12720 arguments).

12721 *LC_MESSAGES*
12722 Determine the locale that should be used to affect the format and contents of
12723 diagnostic messages written to standard error.

12724 XSI NLSPATH Determine the location of message catalogs for the processing of *LC_MESSAGES*.

12725 ASYNCHRONOUS EVENTS

12726 Default.

12727 STDOUT

The **echo** utility arguments shall be separated by single <space>s and a <newline> shall follow the last argument. Output transformations shall occur based on the escape sequences in the input. See the OPERANDS section.

12731 STDERR

12732 The standard error shall be used only for diagnostic messages.

12733 OUTPUT FILES

12734 None.

12735 EXTENDED DESCRIPTION

12736 None.

12737 EXIT STATUS

12738 The following exit values shall be returned:

12739 0 Successful completion.

12740 >0 An error occurred.

12741 CONSEQUENCES OF ERRORS

12742 Default.

12743 APPLICATION USAGE

12744 In the ISO/IEC 9945-2:1993 standard, it was not possible to use *echo* portably across all systems
12745 that were not XSI-conformant unless both **-n** (as the first argument) and escape sequences were
12746 omitted.

12747 The *printf* utility can be used portably to emulate any of the traditional behaviors of the *echo*
12748 utility as follows (assuming that *IFS* has its standard value or is unset):

- The historic System V *echo* and the current requirements in this volume of IEEE Std 1003.1-2001 are equivalent to:

```
12751 printf "%b\n" "$*"
```

- The BSD *echo* is equivalent to:

```
12753     if [ "X$1" = "X-n" ]
12754     then
12755         shift
12756         printf "%s" "$*"
12757     else
12758         printf "%s\n" "$*"
12759     fi
```

12760 New applications are encouraged to use *printf* instead of *echo*.

12761 **EXAMPLES**

12762 None.

12763 **RATIONALE**

12764 The *echo* utility has not been made obsolescent because of its extremely widespread use in
12765 historical applications. Conforming applications that wish to do prompting without <newline>s
12766 or that could possibly be expecting to echo a **-n**, should use the *printf* utility derived from the
12767 Ninth Edition system.

12768 As specified, *echo* writes its arguments in the simplest of ways. The two different historical
12769 versions of *echo* vary in fatally incompatible ways.

12770 The BSD *echo* checks the first argument for the string **-n** which causes it to suppress the
12771 <newline> that would otherwise follow the final argument in the output.

12772 The System V *echo* does not support any options, but allows escape sequences within its
12773 operands, as described in the OPERANDS section.

12774 The *echo* utility does not support Utility Syntax Guideline 10 because historical applications
12775 depend on *echo* to echo *all* of its arguments, except for the **-n** option in the BSD version.

12776 **FUTURE DIRECTIONS**

12777 None.

12778 **SEE ALSO**

12779 *printf*

12780 **CHANGE HISTORY**

12781 First released in Issue 2.

12782 **Issue 5**

12783 In the OPTIONS section, the last sentence is changed to indicate that implementations “do not”
12784 support any options; in the previous issue this said “need not”.

12785 **Issue 6**

12786 The following new requirements on POSIX implementations derive from alignment with the
12787 Single UNIX Specification:

- 12788 • A set of character sequences is defined as *string* operands.
- 12789 • *LC_CTYPE* is added to the list of environment variables affecting *echo*.
- 12790 • In the OPTIONS section, implementations shall not support any options.

12791 NAME

12792 ed — edit text

12793 SYNOPSIS

12794 ed [-p string][-s][file]

12795 DESCRIPTION

12796 The *ed* utility is a line-oriented text editor that uses two modes: *command mode* and *input mode*.
12797 In command mode the input characters shall be interpreted as commands, and in input mode
12798 they shall be interpreted as text. See the EXTENDED DESCRIPTION section.

12799 OPTIONS

12800 The *ed* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
12801 Utility Syntax Guidelines.

12802 The following options shall be supported:

12803 -p *string* Use *string* as the prompt string when in command mode. By default, there shall be
12804 no prompt string.

12805 -s Suppress the writing of byte counts by e, E, r, and w commands and of the '!' prompt after a !command.
12806

12807 OPERANDS

12808 The following operand shall be supported:

12809 file If the *file* argument is given, *ed* shall simulate an e command on the file named by
12810 the pathname, *file*, before accepting commands from the standard input. If the *file*
12811 operand is '−', the results are unspecified.

12812 STDIN

12813 The standard input shall be a text file consisting of commands, as described in the EXTENDED
12814 DESCRIPTION section.

12815 INPUT FILES

12816 The input files shall be text files.

12817 ENVIRONMENT VARIABLES

12818 The following environment variables shall affect the execution of *ed*:

12819 HOME Determine the pathname of the user's home directory.

12820 LANG Provide a default value for the internationalization variables that are unset or null.
12821 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
12822 Internationalization Variables for the precedence of internationalization variables
12823 used to determine the values of locale categories.)

12824 LC_ALL If set to a non-empty string value, override the values of all the other
12825 internationalization variables.

12826 LC_COLLATE

12827 Determine the locale for the behavior of ranges, equivalence classes, and multi-
12828 character collating elements within regular expressions.

12829 LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as
12830 characters (for example, single-byte as opposed to multi-byte characters in
12831 arguments and input files) and the behavior of character classes within regular
12832 expressions.

12833 LC_MESSAGES

12834 Determine the locale that should be used to affect the format and contents of

12835 diagnostic messages written to standard error and informative messages written to
12836 standard output.

12837 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

12838 ASYNCHRONOUS EVENTS

12839 The *ed* utility shall take the standard action for all signals (see the ASYNCHRONOUS EVENTS
12840 section in Section 1.11 (on page 20)) with the following exceptions:

12841 SIGINT The *ed* utility shall interrupt its current activity, write the string "?\n" to standard
12842 output, and return to command mode (see the EXTENDED DESCRIPTION
12843 section).

12844 SIGHUP If the buffer is not empty and has changed since the last write, the *ed* utility shall
12845 attempt to write a copy of the buffer in a file. First, the file named **ed.hup** in the
12846 current directory shall be used; if that fails, the file named **ed.hup** in the directory
12847 named by the *HOME* environment variable shall be used. In any case, the *ed* utility
12848 shall exit without returning to command mode.

12849 SIGQUIT The *ed* utility shall ignore this event.

12850 STDOUT

12851 Various editing commands and the prompting feature (see **-p**) write to standard output, as
12852 described in the EXTENDED DESCRIPTION section.

12853 STDERR

12854 The standard error shall be used only for diagnostic messages.

12855 OUTPUT FILES

12856 The output files shall be text files whose formats are dependent on the editing commands given.

12857 EXTENDED DESCRIPTION

12858 The *ed* utility shall operate on a copy of the file it is editing; changes made to the copy shall have
12859 no effect on the file until a **w** (write) command is given. The copy of the text is called the *buffer*.

12860 Commands to *ed* have a simple and regular structure: zero, one, or two *addresses* followed by a
12861 single-character *command*, possibly followed by parameters to that command. These addresses
12862 specify one or more lines in the buffer. Every command that requires addresses has default
12863 addresses, so that the addresses very often can be omitted. If the **-p** option is specified, the
12864 prompt string shall be written to standard output before each command is read.

12865 In general, only one command can appear on a line. Certain commands allow text to be input.
12866 This text is placed in the appropriate place in the buffer. While *ed* is accepting text, it is said to be
12867 in *input mode*. In this mode, no commands shall be recognized; all input is merely collected.
12868 Input mode is terminated by entering a line consisting of two characters: a period ('.')
12869 followed by a <newline>. This line is not considered part of the input text.

12870 Regular Expressions in ed

12871 The *ed* utility shall support basic regular expressions, as described in the Base Definitions
12872 volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions. Since regular
12873 expressions in *ed* are always matched against single lines (excluding the terminating
12874 <newline>s), never against any larger section of text, there is no way for a regular expression to
12875 match a <newline>.

12876 A null RE shall be equivalent to the last RE encountered.

12877 Regular expressions are used in addresses to specify lines, and in some commands (for example,
12878 the **s** substitute command) to specify portions of a line to be substituted.

12879 **Addresses in ed**

12880 Addressing in *ed* relates to the current line. Generally, the current line is the last line affected by a
12881 command. The current line number is the address of the current line. If the edit buffer is not
12882 empty, the initial value for the current line shall be the last line in the edit buffer; otherwise, zero.

12883 Addresses shall be constructed as follows:

- 12884 1. The period character ('.') shall address the current line.
- 12885 2. The dollar sign character ('\$') shall address the last line of the edit buffer.
- 12886 3. The positive decimal number *n* shall address the *n*th line of the edit buffer.
- 12887 4. The apostrophe-x character pair ("'x") shall address the line marked with the mark name
12888 character *x*, which shall be a lowercase letter from the portable character set. It shall be an
12889 error if the character has not been set to mark a line or if the line that was marked is not
12890 currently present in the edit buffer.
- 12891 5. A BRE enclosed by slash characters ('/') shall address the first line found by searching
12892 forwards from the line following the current line toward the end of the edit buffer and
12893 stopping at the first line for which the line excluding the terminating <newline> matches
12894 the BRE. The BRE consisting of a null BRE delimited by a pair of slash characters shall
12895 address the next line for which the line excluding the terminating <newline> matches the
12896 last BRE encountered. In addition, the second slash can be omitted at the end of a
12897 command line. Within the BRE, a backslash-slash pair ("\//") shall represent a literal slash
12898 instead of the BRE delimiter. If necessary, the search shall wrap around to the beginning of
12899 the buffer and continue up to and including the current line, so that the entire buffer is
12900 searched.
- 12901 6. A BRE enclosed by question-mark characters ('?') shall address the first line found by
12902 searching backwards from the line preceding the current line toward the beginning of the
12903 edit buffer and stopping at the first line for which the line excluding the terminating
12904 <newline> matches the BRE. The BRE consisting of a null BRE delimited by a pair of
12905 question-mark characters ("??") shall address the previous line for which the line
12906 excluding the terminating <newline> matches the last BRE encountered. In addition, the
12907 second question-mark can be omitted at the end of a command line. Within the BRE, a
12908 backslash-question-mark pair ("\?") shall represent a literal question mark instead of the
12909 BRE delimiter. If necessary, the search shall wrap around to the end of the buffer and
12910 continue up to and including the current line, so that the entire buffer is searched.
- 12911 7. A plus-sign ('+') or hyphen character ('-') followed by a decimal number shall address
12912 the current line plus or minus the number. A plus-sign or hyphen character not followed
12913 by a decimal number shall address the current line plus or minus 1.

12914 Addresses can be followed by zero or more address offsets, optionally <blank>-separated.
12915 Address offsets are constructed as follows:

- 12916 • A plus-sign or hyphen character followed by a decimal number shall add or subtract,
12917 respectively, the indicated number of lines to or from the address. A plus-sign or hyphen
12918 character not followed by a decimal number shall add or subtract 1 to or from the address.
- 12919 • A decimal number shall add the indicated number of lines to the address.

12920 It shall not be an error for an intermediate address value to be less than zero or greater than the
12921 last line in the edit buffer. It shall be an error for the final address value to be less than zero or
12922 greater than the last line in the edit buffer. It shall be an error if a search for a BRE fails to find a
12923 matching line.

12924 Commands accept zero, one, or two addresses. If more than the required number of addresses
 12925 are provided to a command that requires zero addresses, it shall be an error. Otherwise, if more
 12926 than the required number of addresses are provided to a command, the addresses specified first
 12927 shall be evaluated and then discarded until the maximum number of valid addresses remain, for
 12928 the specified command.

12929 Addresses shall be separated from each other by a comma (',') or semicolon character (';').
 12930 In the case of a semicolon separator, the current line ('.') shall be set to the first address, and
 12931 only then will the second address be calculated. This feature can be used to determine the
 12932 starting line for forwards and backwards searches; see rules 5. and 6.

12933 Addresses can be omitted on either side of the comma or semicolon separator, in which case the
 12934 resulting address pairs shall be as follows:

Specified	Resulting
,	1 , \$
, addr	1 , addr
addr ,	addr , addr
;	. ; \$
; addr	. ; addr
addr ;	addr ; addr

12942 Any <blank>s included between addresses, address separators, or address offsets shall be
 12943 ignored.

12944 Commands in ed

12945 In the following list of *ed* commands, the default addresses are shown in parentheses. The
 12946 number of addresses shown in the default shall be the number expected by the command. The
 12947 parentheses are not part of the address; they show that the given addresses are the default.

12948 It is generally invalid for more than one command to appear on a line. However, any command
 12949 (except **e**, **E**, **f**, **q**, **Q**, **r**, **w**, and **l**) can be suffixed by the letter **l**, **n**, or **p**; in which case, except for
 12950 the **l**, **n**, and **p** commands, the command shall be executed and then the new current line shall be
 12951 written as described below under the **l**, **n**, and **p** commands. When an **l**, **n**, or **p** suffix is used
 12952 with an **l**, **n**, or **p** command, the command shall write to standard output as described below, but
 12953 it is unspecified whether the suffix writes the current line again in the requested format or
 12954 whether the suffix has no effect. For example, the **pl** command (base **p** command with an **l**
 12955 suffix) shall either write just the current line or write it twice—once as specified for **p** and once
 12956 as specified for **l**. Also, the **g**, **G**, **v**, and **V** commands shall take a command as a parameter.

12957 Each address component can be preceded by zero or more <blank>s. The command letter can be
 12958 preceded by zero or more <blank>s. If a suffix letter (**l**, **n**, or **p**) is given, the application shall
 12959 ensure that it immediately follows the command.

12960 The **e**, **E**, **f**, **r**, and **w** commands shall take an optional *file* parameter, separated from the
 12961 command letter by one or more <blank>s.

12962 If changes have been made in the buffer since the last **w** command that wrote the entire buffer,
 12963 *ed* shall warn the user if an attempt is made to destroy the editor buffer via the **e** or **q** commands.
 12964 The *ed* utility shall write the string:

12965 " ?\n "

12966 (followed by an explanatory message if *help mode* has been enabled via the **H** command) to
 12967 standard output and shall continue in command mode with the current line number unchanged.
 12968 If the **e** or **q** command is repeated with no intervening command, it shall take effect.

12969 If a terminal disconnect is detected:

- 12970 • If the buffer is not empty and has changed since the last write, the *ed* utility shall attempt to
12971 write a copy of the buffer to a file named **ed.hup** in the current directory. If this write fails, *ed*
12972 shall attempt to write a copy of the buffer to a filename **ed.hup** in the directory named by the
12973 *HOME* environment variable. If both these attempts fail, *ed* shall exit without saving the
12974 buffer.
- 12975 • The *ed* utility shall not write the file to the currently remembered pathname or return to
12976 command mode, and shall terminate with a non-zero exit status.

12977 If an end-of-file is detected on standard input:

- 12978 • If the *ed* utility is in input mode, *ed* shall terminate input mode and return to command mode.
12979 It is unspecified if any partially entered lines (that is, input text without a terminating
12980 <newline>) are discarded from the input text.
- 12981 • If the *ed* utility is in command mode, it shall act as if a **q** command had been entered.

12982 If the closing delimiter of an RE or of a replacement string (for example, ' / ') in a **g**, **G**, **s**, **v**, or **V**
12983 command would be the last character before a <newline>, that delimiter can be omitted, in
12984 which case the addressed line shall be written. For example, the following pairs of commands
12985 are equivalent:

12986 s/s1/s2 s/s1/s2/p
12987 g/s1 g/s1/p
12988 ?s1? ?s1?

12989 If an invalid command is entered, *ed* shall write the string:

12990 "?\n"

12991 (followed by an explanatory message if *help mode* has been enabled via the **H** command) to
12992 standard output and shall continue in command mode with the current line number unchanged.

12993 Append Command

12994 *Synopsis:* (.)a
12995 <text>
12996 .

12997 The **a** command shall read the given text and append it after the addressed line; the current line
12998 number shall become the address of the last inserted line or, if there were none, the addressed
12999 line. Address 0 shall be valid for this command; it shall cause the appended text to be placed at
13000 the beginning of the buffer.

13001 Change Command

13002 *Synopsis:* (. . .)c
13003 <text>
13004 .

13005 The **c** command shall delete the addressed lines, then accept input text that replaces these lines;
13006 the current line shall be set to the address of the last line input; or, if there were none, at the line
13007 after the last line deleted; if the lines deleted were originally at the end of the buffer, the current
13008 line number shall be set to the address of the new last line; if no lines remain in the buffer, the
13009 current line number shall be set to zero. Address 0 shall be valid for this command; it shall be
13010 interpreted as if address 1 were specified.

13011 **Delete Command**

13012 *Synopsis:* (. , .)d

13013 The **d** command shall delete the addressed lines from the buffer. The address of the line after the
13014 last line deleted shall become the current line number; if the lines deleted were originally at the
13015 end of the buffer, the current line number shall be set to the address of the new last line; if no
13016 lines remain in the buffer, the current line number shall be set to zero.

13017 **Edit Command**

13018 *Synopsis:* e [file]

13019 The **e** command shall delete the entire contents of the buffer and then read in the file named by
13020 the pathname *file*. The current line number shall be set to the address of the last line of the
13021 buffer. If no pathname is given, the currently remembered pathname, if any, shall be used (see
13022 the **f** command). The number of bytes read shall be written to standard output, unless the **-s**
13023 option was specified, in the following format:

13024 "%d\n" , <number of bytes read>

13025 The name *file* shall be remembered for possible use as a default pathname in subsequent **e**, **E**, **r**,
13026 and **w** commands. If *file* is replaced by '!', the rest of the line shall be taken to be a shell
13027 command line whose output is to be read. Such a shell command line shall not be remembered
13028 as the current *file*. All marks shall be discarded upon the completion of a successful **e** command.
13029 If the buffer has changed since the last time the entire buffer was written, the user shall be
13030 warned, as described previously.

13031 **Edit Without Checking Command**

13032 *Synopsis:* E [file]

13033 The **E** command shall possess all properties and restrictions of the **e** command except that the
13034 editor shall not check to see whether any changes have been made to the buffer since the last **w**
13035 command.

13036 **Filename Command**

13037 *Synopsis:* f [file]

13038 If *file* is given, the **f** command shall change the currently remembered pathname to *file*; whether
13039 the name is changed or not, it shall then write the (possibly new) currently remembered
13040 pathname to the standard output in the following format:

13041 "%s\n" , <pathname>

13042 The current line number shall be unchanged.

13043 **Global Command**

13044 *Synopsis:* (1,\$)g/RE/command list

13045 In the **g** command, the first step shall be to mark every line for which the line excluding the
13046 terminating <newline> matches the given RE. Then, going sequentially from the beginning of
13047 the file to the end of the file, the given *command list* shall be executed for each marked line, with
13048 the current line number set to the address of that line. Any line modified by the *command list*
13049 shall be unmarked. When the **g** command completes, the current line number shall have the
13050 value assigned by the last command in the *command list*. If there were no matching lines, the
13051 current line number shall not be changed. A single command or the first of a list of commands

13052 shall appear on the same line as the global command. All lines of a multi-line list except the last
13053 line shall be ended with a backslash preceding the terminating <newline>; the **a**, **i**, and **c**
13054 commands and associated input are permitted. The ' . ' terminating input mode can be omitted
13055 if it would be the last line of the *command list*. An empty *command list* shall be equivalent to the **p**
13056 command. The use of the **g**, **G**, **v**, **V**, and **!** commands in the *command list* produces undefined
13057 results. Any character other than <space> or <newline> can be used instead of a slash to delimit
13058 the RE. Within the RE, the RE delimiter itself can be used as a literal character if it is preceded by
13059 a backslash.

13060 Interactive Global Command

13061 *Synopsis:* (1 , \$)G/RE/

13062 In the **G** command, the first step shall be to mark every line for which the line excluding the
13063 terminating <newline> matches the given RE. Then, for every such line, that line shall be
13064 written, the current line number shall be set to the address of that line, and any one command
13065 (other than one of the **a**, **c**, **i**, **g**, **G**, **v**, and **V** commands) shall be read and executed. A <newline>
13066 shall act as a null command (causing no action to be taken on the current line); an '&' shall
13067 cause the re-execution of the most recent non-null command executed within the current
13068 invocation of **G**. Note that the commands input as part of the execution of the **G** command can
13069 address and affect any lines in the buffer. The final value of the current line number shall be the
13070 value set by the last command successfully executed. (Note that the last command successfully
13071 executed shall be the **G** command itself if a command fails or the null command is specified.) If
13072 there were no matching lines, the current line number shall not be changed. The **G** command can
13073 be terminated by a SIGINT signal. Any character other than <space> or <newline> can be used
13074 instead of a slash to delimit the RE and the replacement. Within the RE, the RE delimiter itself
13075 can be used as a literal character if it is preceded by a backslash.

13076 Help Command

13077 *Synopsis:* h

13078 The **h** command shall write a short message to standard output that explains the reason for the
13079 most recent '?' notification. The current line number shall be unchanged.

13080 Help-Mode Command

13081 *Synopsis:* H

13082 The **H** command shall cause *ed* to enter a mode in which help messages (see the **h** command)
13083 shall be written to standard output for all subsequent '?' notifications. The **H** command
13084 alternately shall turn this mode on and off; it is initially off. If the help-mode is being turned on,
13085 the **H** command also explains the previous '?' notification, if there was one. The current line
13086 number shall be unchanged.

13087 Insert Command

13088 *Synopsis:* (.)i
13089 <text>
13090 .

13091 The **i** command shall insert the given text before the addressed line; the current line is set to the
13092 last inserted line or, if there was none, to the addressed line. This command differs from the **a**
13093 command only in the placement of the input text. Address 0 shall be valid for this command; it
13094 shall be interpreted as if address 1 were specified.

13095 **Join Command**

13096 *Synopsis:* (. , . +1) j

13097 The **j** command shall join contiguous lines by removing the appropriate <newline>s. If exactly
13098 one address is given, this command shall do nothing. If lines are joined, the current line number
13099 shall be set to the address of the joined line; otherwise, the current line number shall be
13100 unchanged.

13101 **Mark Command**

13102 *Synopsis:* (.) kx

13103 The **k** command shall mark the addressed line with name *x*, which the application shall ensure is
13104 a lowercase letter from the portable character set. The address "'x'" shall then refer to this line;
13105 the current line number shall be unchanged.

13106 **List Command**

13107 *Synopsis:* (. , .) l

13108 The **l** command shall write to standard output the addressed lines in a visually unambiguous
13109 form. The characters listed in the Base Definitions volume of IEEE Std 1003.1-2001, Table 5-1,
13110 Escape Sequences and Associated Actions ('\\', '\\a', '\\b', '\\f', '\\r', '\\t', '\\v') shall
13111 be written as the corresponding escape sequence; the '\\n' in that table is not applicable. Non-
13112 printable characters not in the table shall be written as one three-digit octal number (with a
13113 preceding backslash character) for each byte in the character (most significant byte first). If the
13114 size of a byte on the system is greater than nine bits, the format used for non-printable characters
13115 is implementation-defined.

13116 Long lines shall be folded, with the point of folding indicated by <newline> preceded by a
13117 backslash; the length at which folding occurs is unspecified, but should be appropriate for the
13118 output device. The end of each line shall be marked with a '\$', and '\$' characters within the
13119 text shall be written with a preceding backslash. An **l** command can be appended to any other
13120 command other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**. The current line number shall be set to the address of
13121 the last line written.

13122 **Move Command**

13123 *Synopsis:* (. , .) maddress

13124 The **m** command shall reposition the addressed lines after the line addressed by *address*.
13125 Address 0 shall be valid for *address* and cause the addressed lines to be moved to the beginning
13126 of the buffer. It shall be an error if address *address* falls within the range of moved lines. The
13127 current line number shall be set to the address of the last line moved.

13128 **Number Command**

13129 *Synopsis:* (. , .) n

13130 The **n** command shall write to standard output the addressed lines, preceding each line by its
13131 line number and a <tab>; the current line number shall be set to the address of the last line
13132 written. The **n** command can be appended to any command other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**.

13133 **Print Command**13134 *Synopsis:* (. . .)**p**

13135 The **p** command shall write to standard output the addressed lines; the current line number shall
13136 be set to the address of the last line written. The **p** command can be appended to any command
13137 other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**.

13138 **Prompt Command**13139 *Synopsis:* **P**

13140 The **P** command shall cause *ed* to prompt with an asterisk (' * ') (or *string*, if **-p** is specified) for
13141 all subsequent commands. The **P** command alternatively shall turn this mode on and off; it shall
13142 be initially on if the **-p** option is specified; otherwise, off. The current line number shall be
13143 unchanged.

13144 **Quit Command**13145 *Synopsis:* **q**

13146 The **q** command shall cause *ed* to exit. If the buffer has changed since the last time the entire
13147 buffer was written, the user shall be warned, as described previously.

13148 **Quit Without Checking Command**13149 *Synopsis:* **Q**

13150 The **Q** command shall cause *ed* to exit without checking whether changes have been made in the
13151 buffer since the last **w** command.

13152 **Read Command**13153 *Synopsis:* (\$)**r** [*file*]

13154 The **r** command shall read in the file named by the pathname *file* and append it after the
13155 addressed line. If no *file* argument is given, the currently remembered pathname, if any, shall be
13156 used (see the **e** and **f** commands). The currently remembered pathname shall not be changed
13157 unless there is no remembered pathname. Address 0 shall be valid for **r** and shall cause the file to
13158 be read at the beginning of the buffer. If the read is successful, and **-s** was not specified, the
13159 number of bytes read shall be written to standard output in the following format:

13160 "%d\n", <number of bytes read>

13161 The current line number shall be set to the address of the last line read in. If *file* is replaced by
13162 ' ! ', the rest of the line shall be taken to be a shell command line whose output is to be read.
13163 Such a shell command line shall not be remembered as the current pathname.

13164 **Substitute Command**13165 *Synopsis:* (. . .)**s**/*RE/replacement/flags*

13166 The **s** command shall search each addressed line for an occurrence of the specified *RE* and
13167 replace either the first or all (non-overlapped) matched strings with the *replacement*; see the
13168 following description of the **g** suffix. It is an error if the substitution fails on every addressed
13169 line. Any character other than <space> or <newline> can be used instead of a slash to delimit the
13170 *RE* and the *replacement*. Within the *RE*, the *RE* delimiter itself can be used as a literal character
13171 if it is preceded by a backslash. The current line shall be set to the address of the last line on
13172 which a substitution occurred.

13173 An ampersand ('&') appearing in the *replacement* shall be replaced by the string matching the
13174 RE on the current line. The special meaning of '&' in this context can be suppressed by
13175 preceding it by backslash. As a more general feature, the characters '\n', where *n* is a digit,
13176 shall be replaced by the text matched by the corresponding back-reference expression. When the
13177 character '%' is the only character in the *replacement*, the *replacement* used in the most recent
13178 substitute command shall be used as the *replacement* in the current substitute command; if there
13179 was no previous substitute command, the use of '%' in this manner shall be an error. The '%'
13180 shall lose its special meaning when it is in a replacement string of more than one character or is
13181 preceded by a backslash. For each backslash ('\') encountered in scanning *replacement* from
13182 beginning to end, the following character shall lose its special meaning (if any). It is unspecified
13183 what special meaning is given to any character other than '&', '\', '%', or digits.

13184 A line can be split by substituting a <newline> into it. The application shall ensure it escapes the
13185 <newline> in the *replacement* by preceding it by backslash. Such substitution cannot be done as
13186 part of a **g** or **v** *command list*. The current line number shall be set to the address of the last line
13187 on which a substitution is performed. If no substitution is performed, the current line number
13188 shall be unchanged. If a line is split, a substitution shall be considered to have been performed
13189 on each of the new lines for the purpose of determining the new current line number. A
13190 substitution shall be considered to have been performed even if the replacement string is
13191 identical to the string that it replaces.

13192 The application shall ensure that the value of *flags* is zero or more of:

- 13193 **count** Substitute for the *count*th occurrence only of the RE found on each addressed line.
- 13194 **g** Globally substitute for all non-overlapping instances of the RE rather than just the first
13195 one. If both **g** and **count** are specified, the results are unspecified.
- 13196 **I** Write to standard output the final line in which a substitution was made. The line shall
13197 be written in the format specified for the **I** command.
- 13198 **n** Write to standard output the final line in which a substitution was made. The line shall
13199 be written in the format specified for the **n** command.
- 13200 **p** Write to standard output the final line in which a substitution was made. The line shall
13201 be written in the format specified for the **p** command.

13202 **Copy Command**

13203 *Synopsis:* (. . .)taddress

13204 The **t** command shall be equivalent to the **m** command, except that a copy of the addressed lines
13205 shall be placed after address *address* (which can be 0); the current line number shall be set to the
13206 address of the last line added.

13207 **Undo Command**

13208 *Synopsis:* u

13209 The **u** command shall nullify the effect of the most recent command that modified anything in
13210 the buffer, namely the most recent **a**, **c**, **d**, **g**, **i**, **j**, **m**, **r**, **s**, **t**, **u**, **v**, **G**, or **V** command. All changes
13211 made to the buffer by a **g**, **G**, **v**, or **V** global command shall be undone as a single change; if no
13212 changes were made by the global command (such as with **g/RE/p**), the **u** command shall have
13213 no effect. The current line number shall be set to the value it had immediately before the
13214 command being undone started.

13215 **Global Non-Matched Command**

13216 *Synopsis:* (1,\$)v/*RE*/command list

13217 This command shall be equivalent to the global command g except that the lines that are marked
13218 during the first step shall be those for which the line excluding the terminating <newline> does
13219 not match the RE.

13220 **Interactive Global Not-Matched Command**

13221 *Synopsis:* (1,\$)V/*RE*/

13222 This command shall be equivalent to the interactive global command G except that the lines that
13223 are marked during the first step shall be those for which the line excluding the terminating
13224 <newline> does not match the RE.

13225 **Write Command**

13226 *Synopsis:* (1,\$)w [*file*]

13227 The w command shall write the addressed lines into the file named by the pathname *file*. The
13228 command shall create the file, if it does not exist, or shall replace the contents of the existing file.
13229 The currently remembered pathname shall not be changed unless there is no remembered
13230 pathname. If no pathname is given, the currently remembered pathname, if any, shall be used
13231 (see the e and f commands); the current line number shall be unchanged. If the command is
13232 successful, the number of bytes written shall be written to standard output, unless the -s option
13233 was specified, in the following format:

13234 "%d\n", <number of bytes written>

13235 If *file* begins with '!', the rest of the line shall be taken to be a shell command line whose
13236 standard input shall be the addressed lines. Such a shell command line shall not be remembered
13237 as the current pathname. This usage of the write command with '!' shall not be considered as a
13238 "last w command that wrote the entire buffer", as described previously; thus, this alone shall not
13239 prevent the warning to the user if an attempt is made to destroy the editor buffer via the e or q
13240 commands.

13241 **Line Number Command**

13242 *Synopsis:* (\$)=

13243 The line number of the addressed line shall be written to standard output in the following
13244 format:

13245 "%d\n", <line number>

13246 The current line number shall be unchanged by this command.

13247 **Shell Escape Command**

13248 *Synopsis:* !*command*

13249 The remainder of the line after the '!' shall be sent to the command interpreter to be
13250 interpreted as a shell command line. Within the text of that shell command line, the unescaped
13251 character '%' shall be replaced with the remembered pathname; if a '!' appears as the first
13252 character of the command, it shall be replaced with the text of the previous shell command
13253 executed via '!'. Thus, " !!" shall repeat the previous !*command*. If any replacements of '%' or
13254 '!' are performed, the modified line shall be written to the standard output before *command* is
13255 executed. The ! command shall write:

13256 "!\\n"
13257 to standard output upon completion, unless the **-s** option is specified. The current line number
13258 shall be unchanged.

13259 **Null Command**

13260 *Synopsis:* (.+1)

13261 An address alone on a line shall cause the addressed line to be written. A <newline> alone shall
13262 be equivalent to "+1p". The current line number shall be set to the address of the written line.

13263 **EXIT STATUS**

13264 The following exit values shall be returned:

13265 0 Successful completion without any file or command errors.

13266 >0 An error occurred.

13267 **CONSEQUENCES OF ERRORS**

13268 When an error in the input script is encountered, or when an error is detected that is a
13269 consequence of the data (not) present in the file or due to an external condition such as a read or
13270 write error:

- 13271 • If the standard input is a terminal device file, all input shall be flushed, and a new command
13272 read.
- 13273 • If the standard input is a regular file, *ed* shall terminate with a non-zero exit status.

13274 **APPLICATION USAGE**

13275 Because of the extremely terse nature of the default error messages, the prudent script writer
13276 begins the *ed* input commands with an **H** command, so that if any errors do occur at least some
13277 clue as to the cause is made available.

13278 In previous versions, an obsolescent – option was described. This is no longer specified.
13279 Applications should use the **-s** option. Using – as a *file* operand now produces unspecified
13280 results. This allows implementations to continue to support the former required behavior.

13281 **EXAMPLES**

13282 None.

13283 **RATIONALE**

13284 The initial description of this utility was adapted from the SVID. It contains some features not
13285 found in Version 7 or BSD-derived systems. Some of the differences between the POSIX and
13286 BSD *ed* utilities include, but need not be limited to:

- 13287 • The BSD – option does not suppress the '!' prompt after a ! command.
- 13288 • BSD does not support the special meanings of the '%' and '!' characters within a !
13289 command.
- 13290 • BSD does not support the addresses ';' and ',', '.
- 13291 • BSD allows the command/suffix pairs **pp**, **ll**, and so on, which are unspecified in this volume
13292 of IEEE Std 1003.1-2001.
- 13293 • BSD does not support the '!' character part of the **e**, **r**, or **w** commands.
- 13294 • A failed **g** command in BSD sets the line number to the last line searched if there are no
13295 matches.

- 13296 • BSD does not default the *command list* to the **p** command.
- 13297 • BSD does not support the **G**, **h**, **H**, **n**, or **V** commands.
- 13298 • On BSD, if there is no inserted text, the *insert* command changes the current line to the
13299 referenced line -1; that is, the line before the specified line.
- 13300 • On BSD, the *join* command with only a single address changes the current line to that
13301 address.
- 13302 • BSD does not support the **P** command; moreover, in BSD it is synonymous with the **p**
13303 command.
- 13304 • BSD does not support the *undo* of the commands **j**, **m**, **r**, **s**, or **t**.
- 13305 • The Version 7 *ed* command **W**, and the BSD *ed* commands **W**, **wq**, and **z** are not present in this
13306 volume of IEEE Std 1003.1-2001.

13307 The **-s** option was added to allow the functionality of the now withdrawn **-** option in a manner
13308 compatible with the Utility Syntax Guidelines.

13309 In early proposals there was a limit, {ED_FILE_MAX}, that described the historical limitations of
13310 some *ed* utilities in their handling of large files; some of these have had problems with files larger
13311 than 100 000 bytes. It was this limitation that prompted much of the desire to include a *split*
13312 command in this volume of IEEE Std 1003.1-2001. Since this limit was removed, this volume of
13313 IEEE Std 1003.1-2001 requires that implementations document the file size limits imposed by *ed*
13314 in the conformance document. The limit {ED_LINE_MAX} was also removed; therefore, the
13315 global limit {LINE_MAX} is used for input and output lines.

13316 The manner in which the **I** command writes non-printable characters was changed to avoid the
13317 historical backspace-overstrike method. On video display terminals, the overstrike is ambiguous
13318 because most terminals simply replace overstruck characters, making the **I** format not useful for
13319 its intended purpose of unambiguously understanding the content of the line. The historical
13320 backslash escapes were also ambiguous. (The string "a\0011" could represent a line containing
13321 those six characters or a line containing the three characters 'a', a byte with a binary value of 1,
13322 and a 1.) In the format required here, a backslash appearing in the line is written as "\\" so that
13323 the output is truly unambiguous. The method of marking the ends of lines was adopted from the
13324 *ex* editor and is required for any line ending in <space>s; the '\$' is placed on all lines so that a
13325 real '\$' at the end of a line cannot be misinterpreted.

13326 Systems with bytes too large to fit into three octal digits must devise other means of displaying
13327 non-printable characters. Consideration was given to requiring that the number of octal digits be
13328 large enough to hold a byte, but this seemed to be too confusing for applications on the vast
13329 majority of systems where three digits are adequate. It would be theoretically possible for the
13330 application to use the *getconf* utility to find out the CHAR_BIT value and deal with such an
13331 algorithm; however, there is really no portable way that an application can use the octal values
13332 of the bytes across various coded character sets, so the additional specification was not
13333 worthwhile.

13334 The description of how a NUL is written was removed. The NUL character cannot be in text
13335 files, and this volume of IEEE Std 1003.1-2001 should not dictate behavior in the case of
13336 undefined, erroneous input.

13337 Unlike some of the other editing utilities, the filenames accepted by the **E**, **e**, **R**, and **r** commands
13338 are not patterns.

13339 Early proposals stated that the **-p** option worked only when standard input was associated with
13340 a terminal device. This has been changed to conform to historical implementations, thereby
13341 allowing applications to interpose themselves between a user and the *ed* utility.

13342 The form of the substitute command that uses the **n** suffix was limited in some historical
13343 documentation (where this was described incorrectly as “backreferencing”). This limit has been
13344 omitted because there is no reason why an editor processing lines of {LINE_MAX} length should
13345 have this restriction. The command **s/x/X/2047** should be able to substitute the 2047th
13346 occurrence of ‘x’ on a line.

13347 The use of printing commands with printing suffixes (such as **pn**, **lp**, and so on) was made
13348 unspecified because BSD-based systems allow this, whereas System V does not.

13349 Some BSD-based systems exit immediately upon receipt of end-of-file if all of the lines in the file
13350 have been deleted. Since this volume of IEEE Std 1003.1-2001 refers to the **q** command in this
13351 instance, such behavior is not allowed.

13352 Some historical implementations returned exit status zero even if command errors had occurred;
13353 this is not allowed by this volume of IEEE Std 1003.1-2001.

13354 Some historical implementations contained a bug that allowed a single period to be entered in
13355 input mode as <backslash> <period> <newline>. This is not allowed by **ed** because there is no
13356 description of escaping any of the characters in input mode; backslashes are entered into the
13357 buffer exactly as typed. The typical method of entering a single period has been to precede it
13358 with another character and then use the substitute command to delete that character.

13359 It is difficult under some modes of some versions of historical operating system terminal drivers
13360 to distinguish between an end-of-file condition and terminal disconnect. IEEE Std 1003.1-2001
13361 does not require implementations to distinguish between the two situations, which permits
13362 historical implementations of the **ed** utility on historical platforms to conform. Implementations
13363 are encouraged to distinguish between the two, if possible, and take appropriate action on
13364 terminal disconnect.

13365 Historically, **ed** accepted a zero address for the **a** and **r** commands in order to insert text at the
13366 start of the edit buffer. When the buffer was empty the command **.=** returned zero.
13367 IEEE Std 1003.1-2001 requires conformance to historical practice.

13368 For consistency with the **a** and **r** commands and better user functionality, the **i** and **c** commands
13369 must also accept an address of 0, in which case **0i** is treated as **1i** and likewise for the **c**
13370 command.

13371 All of the following are valid addresses:

13372 **+++** Three lines after the current line.

13373 **/pattern/-** One line before the next occurrence of pattern.

13374 **-2** Two lines before the current line.

13375 **3 ----- 2** Line one (note the intermediate negative address).

13376 **1 2 3** Line six.

13377 Any number of addresses can be provided to commands taking addresses; for example,
13378 "**1 , 2 , 3 , 4 , 5p**" prints lines 4 and 5, because two is the greatest valid number of addresses
13379 accepted by the **print** command. This, in combination with the semicolon delimiter, permits
13380 users to create commands based on ordered patterns in the file. For example, the command
13381 "**3 ; /foo/ ; +2p**" will display the first line after line 3 that contains the pattern **foo**, plus the next
13382 two lines. Note that the address "**3 ;**" must still be evaluated before being discarded, because
13383 the search origin for the "**/foo/**" command depends on this.

13384 Historically, **ed** disallowed address chains, as discussed above, consisting solely of comma or
13385 semicolon separators; for example, "**, , ,**" or "**; ; ;**" were considered an error. For consistency of
13386 address specification, this restriction is removed. The following table lists some of the address

13387 forms now possible:

13388	Address	Addr1	Addr2	Status	Comment
13389	7 ,	7	7	Historical	
13390	7 , 5 ,	5	5	Historical	
13391	7 , 5 , 9	5	9	Historical	
13392	7 , 9	7	9	Historical	
13393	7 , +	7	8	Historical	
13394	,	1	\$	Historical	
13395	, 7	1	7	Extension	
13396	, ,	\$	\$	Extension	
13397	, ;	\$	\$	Extension	
13398	7 ;	7	7	Historical	
13399	7 ; 5 ;	5	5	Historical	
13400	7 ; 5 ; 9	5	9	Historical	
13401	7 ; 5 , 9	5	9	Historical	
13402	7 ; \$; 4	\$	4	Historical	Valid, but erroneous.
13403	7 ; 9	7	9	Historical	
13404	7 ; +	7	8	Historical	
13405	7 ;	.	\$	Historical	
13406	; 7	.	7	Extension	
13407	; ;	\$	\$	Extension	
13408	; ,	\$	\$	Extension	

13409 Historically, values could be added to addresses by including them after one or more <blank>s;
 13410 for example, "3 – 5p" wrote the seventh line of the file, and "/foo/ 5" was the same as
 13411 "5 /foo/". However, only absolute values could be added; for example, "5 /foo/" was an
 13412 error. IEEE Std 1003.1-2001 requires conformance to historical practice.

13413 Historically, *ed* accepted the '^' character as an address, in which case it was identical to the
 13414 hyphen character. IEEE Std 1003.1-2001 does not require or prohibit this behavior.

13415 FUTURE DIRECTIONS

13416 None.

13417 SEE ALSO

13418 Section 1.11 (on page 20), *ex*, *sed*, *sh*, *vi*

13419 CHANGE HISTORY

13420 First released in Issue 2.

13421 Issue 5

13422 In the OPTIONS section, the meaning of –s and – is clarified.

13423 A second FUTURE DIRECTION is added.

13424 Issue 6

13425 The obsolescent single-minus form is removed.

13426 A second APPLICATION USAGE note is added.

13427 The Open Group Corrigendum U025/2 is applied, correcting the description of the Edit section.

13428 The *ed* utility is updated to align with the IEEE P1003.2b draft standard. This includes addition of
 13429 the treatment of the SIGQUIT signal, changes to *ed* addressing, and changes to processing when
 13430 end-of-file is detected and when terminal disconnect is detected.

13431

The normative text is reworded to avoid use of the term “must” for application requirements.

13432 NAME

13433 env — set the environment for command invocation

13434 SYNOPSIS

13435 env [-i][name=value]... [utility [argument...]]

13436 DESCRIPTION

13437 The *env* utility shall obtain the current environment, modify it according to its arguments, then
13438 invoke the utility named by the *utility* operand with the modified environment.

13439 Optional arguments shall be passed to *utility*.

13440 If no *utility* operand is specified, the resulting environment shall be written to the standard
13441 output, with one *name=value* pair per line.

13442 OPTIONS

13443 The *env* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
13444 12.2, Utility Syntax Guidelines.

13445 The following options shall be supported:

13446 **-i** Invoke *utility* with exactly the environment specified by the arguments; the
13447 inherited environment shall be ignored completely.

13448 OPERANDS

13449 The following operands shall be supported:

13450 *name=value* Arguments of the form *name=value* shall modify the execution environment, and
13451 shall be placed into the inherited environment before the *utility* is invoked.

13452 *utility* The name of the utility to be invoked. If the *utility* operand names any of the
13453 special built-in utilities in Section 2.14 (on page 64), the results are undefined.

13454 *argument* A string to pass as an argument for the invoked utility.

13455 STDIN

13456 Not used.

13457 INPUT FILES

13458 None.

13459 ENVIRONMENT VARIABLES

13460 The following environment variables shall affect the execution of *env*:

13461 *LANG* Provide a default value for the internationalization variables that are unset or null.
13462 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
13463 Internationalization Variables for the precedence of internationalization variables
13464 used to determine the values of locale categories.)

13465 *LC_ALL* If set to a non-empty string value, override the values of all the other
13466 internationalization variables.

13467 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
13468 characters (for example, single-byte as opposed to multi-byte characters in
13469 arguments).

13470 *LC_MESSAGES*

13471 Determine the locale that should be used to affect the format and contents of
13472 diagnostic messages written to standard error.

13473 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

13474 **PATH** Determine the location of the *utility*, as described in the Base Definitions volume of
13475 IEEE Std 1003.1-2001, Chapter 8, Environment Variables. If *PATH* is specified as a
13476 *name=value* operand to *env*, the *value* given shall be used in the search for *utility*.

13477 ASYNCHRONOUS EVENTS

13478 Default.

13479 STDOUT

13480 If no *utility* operand is specified, each *name=value* pair in the resulting environment shall be
13481 written in the form:

13482 "%s=%s\n", <*name*>, <*value*>

13483 If the *utility* operand is specified, the *env* utility shall not write to standard output.

13484 STDERR

13485 The standard error shall be used only for diagnostic messages.

13486 OUTPUT FILES

13487 None.

13488 EXTENDED DESCRIPTION

13489 None.

13490 EXIT STATUS

13491 If *utility* is invoked, the exit status of *env* shall be the exit status of *utility*; otherwise, the *env*
13492 utility shall exit with one of the following values:

13493 0 The *env* utility completed successfully.

13494 1–125 An error occurred in the *env* utility.

13495 126 The utility specified by *utility* was found but could not be invoked.

13496 127 The utility specified by *utility* could not be found.

13497 CONSEQUENCES OF ERRORS

13498 Default.

13499 APPLICATION USAGE

13500 The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if
13501 an error occurs so that applications can distinguish “failure to find a utility” from “invoked
13502 utility exited with an error indication”. The value 127 was chosen because it is not commonly
13503 used for other meanings; most utilities use small values for “normal error conditions” and the
13504 values above 128 can be confused with termination due to receipt of a signal. The value 126 was
13505 chosen in a similar manner to indicate that the utility could be found, but not invoked. Some
13506 scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction
13507 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to
13508 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for
13509 any other reason.

13510 Historical implementations of the *env* utility use the *execvp()* or *execvp()* functions defined in the
13511 System Interfaces volume of IEEE Std 1003.1-2001 to invoke the specified utility; this provides
13512 better performance and keeps users from having to escape characters with special meaning to
13513 the shell. Therefore, shell functions, special built-ins, and built-ins that are only provided by the
13514 shell are not found.

13515 EXAMPLES

13516 The following command:

13517 `env -i PATH=/mybin mygrep xyz myfile`

13518 invokes the command *mygrep* with a new *PATH* value as the only entry in its environment. In
13519 this case, *PATH* is used to locate *mygrep*, which then must reside in **/mybin**.

13520 RATIONALE

13521 As with all other utilities that invoke other utilities, this volume of IEEE Std 1003.1-2001 only
13522 specifies what *env* does with standard input, standard output, standard error, input files, and
13523 output files. If a utility is executed, it is not constrained by the specification of input and output
13524 by *env*.

13525 The **-i** option was added to allow the functionality of the withdrawn – option in a manner
13526 compatible with the Utility Syntax Guidelines.

13527 Some have suggested that *env* is redundant since the same effect is achieved by:

13528 `name=value ... utility [argument ...]`

13529 The example is equivalent to *env* when an environment variable is being added to the
13530 environment of the command, but not when the environment is being set to the given value.
13531 The *env* utility also writes out the current environment if invoked without arguments. There is
13532 sufficient functionality beyond what the example provides to justify inclusion of *env*.

13533 FUTURE DIRECTIONS

13534 None.

13535 SEE ALSO

13536 Section 2.5 (on page 33), Section 2.14 (on page 64)

13537 CHANGE HISTORY

13538 First released in Issue 2.

13539 NAME

13540 ex — text editor

13541 SYNOPSIS

13542 UP ex [-rR][-s | -v][-c command][-t tagstring][-w size][file ...]

13543

13544 DESCRIPTION

13545 The *ex* utility is a line-oriented text editor. There are two other modes of the editor—open and
13546 visual—in which screen-oriented editing is available. This is described more fully by the *ex open*
13547 and *visual* commands and in *vi*.

13548 This section uses the term *edit buffer* to describe the current working text. No specific
13549 implementation is implied by this term. All editing changes are performed on the edit buffer,
13550 and no changes to it shall affect any file until an editor command writes the file.

13551 Certain terminals do not have all the capabilities necessary to support the complete *ex* definition,
13552 such as the full-screen editing commands (*visual mode* or *open mode*). When these commands
13553 cannot be supported on such terminals, this condition shall not produce an error message such
13554 as “not an editor command” or report a syntax error. The implementation may either accept the
13555 commands and produce results on the screen that are the result of an unsuccessful attempt to
13556 meet the requirements of this volume of IEEE Std 1003.1-2001 or report an error describing the
13557 terminal-related deficiency.

13558 OPTIONS

13559 The *ex* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
13560 Utility Syntax Guidelines.

13561 The following options shall be supported:

13562 **-c command** Specify an initial command to be executed in the first edit buffer loaded from an
13563 existing file (see the EXTENDED DESCRIPTION section). Implementations may
13564 support more than a single **-c** option. In such implementations, the specified
13565 commands shall be executed in the order specified on the command line.

13566 **-r** Recover the named files (see the EXTENDED DESCRIPTION section). Recovery
13567 information for a file shall be saved during an editor or system crash (for example,
13568 when the editor is terminated by a signal which the editor can catch), or after the
13569 use of an *ex preserve* command.

13570 A *crash* in this context is an unexpected failure of the system or utility that requires
13571 restarting the failed system or utility. A system crash implies that any utilities
13572 running at the time also crash. In the case of an editor or system crash, the number
13573 of changes to the edit buffer (since the most recent **preserve** command) that will be
13574 recovered is unspecified.

13575 If no *file* operands are given and the **-t** option is not specified, all other options, the
13576 *EXINIT* variable, and any *.exrc* files shall be ignored; a list of all recoverable files
13577 available to the invoking user shall be written, and the editor shall exit normally
13578 without further action.

13579 **-R** Set **readonly** edit option.

13580 **-s** Prepare *ex* for batch use by taking the following actions:

- 13581 • Suppress writing prompts and informational (but not diagnostic) messages.
- 13582 • Ignore the value of *TERM* and any implementation default terminal type and
13583 assume the terminal is a type incapable of supporting open or visual modes;

- 13584 see the **visual** command and the description of *vi*.
- 13585 • Suppress the use of the *EXINIT* environment variable and the reading of any
13586 *.exrc* file; see the EXTENDED DESCRIPTION section.
- 13587 • Suppress autoindentation, ignoring the value of the **autoindent** edit option.
- 13588 **-t tagstring** Edit the file containing the specified *tagstring*; see *ctags*. The tags feature
13589 represented by **-t tagstring** and the **tag** command is optional. It shall be provided
13590 on any system that also provides a conforming implementation of *ctags*; otherwise,
13591 the use of **-t** produces undefined results. On any system, it shall be an error to
13592 specify more than a single **-t** option.
- 13593 **-v** Begin in visual mode (see *vi*).
- 13594 **-w size** Set the value of the *window* editor option to *size*.

13595 OPERANDS

13596 The following operand shall be supported:

13597 *file* A pathname of a file to be edited.

13598 STDIN

13599 The standard input consists of a series of commands and input text, as described in the
13600 EXTENDED DESCRIPTION section. The implementation may limit each line of standard input
13601 to a length of {LINE_MAX}.

13602 If the standard input is not a terminal device, it shall be as if the **-s** option had been specified.

13603 If a read from the standard input returns an error, or if the editor detects an end-of-file condition
13604 from the standard input, it shall be equivalent to a SIGHUP asynchronous event.

13605 INPUT FILES

13606 Input files shall be text files or files that would be text files except for an incomplete last line that
13607 is not longer than {LINE_MAX}-1 bytes in length and contains no NUL characters. By default,
13608 any incomplete last line shall be treated as if it had a trailing <newline>. The editing of other
13609 forms of files may optionally be allowed by *ex* implementations.

13610 The *.exrc* files and source files shall be text files consisting of *ex* commands; see the EXTENDED
13611 DESCRIPTION section.

13612 By default, the editor shall read lines from the files to be edited without interpreting any of those
13613 lines as any form of editor command.

13614 ENVIRONMENT VARIABLES

13615 The following environment variables shall affect the execution of *ex*:

13616 **COLUMNS** Override the system-selected horizontal screen size. See the Base Definitions
13617 volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables for valid values
13618 and results when it is unset or null.

13619 **EXINIT** Determine a list of *ex* commands that are executed on editor start-up. See the
13620 EXTENDED DESCRIPTION section for more details of the initialization phase.

13621 **HOME** Determine a pathname of a directory that shall be searched for an editor start-up
13622 file named *.exrc*; see the EXTENDED DESCRIPTION section.

13623 **LANG** Provide a default value for the internationalization variables that are unset or null.
13624 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
13625 Internationalization Variables for the precedence of internationalization variables
13626 used to determine the values of locale categories.)

13627	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
13628		
13629	<i>LC_COLLATE</i>	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.
13630		
13631		
13632	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), the behavior of character classes within regular expressions, the classification of characters as uppercase or lowercase letters, the case conversion of letters, and the detection of word boundaries.
13633		
13634		
13635		
13636		
13637	<i>LC_MESSAGES</i>	
13638		Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
13639		
13640	<i>LINES</i>	Override the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables for valid values and results when it is unset or null.
13641		
13642		
13643		
13644 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
13645	<i>PATH</i>	Determine the search path for the shell command specified in the <i>ex</i> editor commands ! , shell , read , and write , and the open and visual mode command ! ; see the description of command search and execution in Section 2.9.1.1 (on page 48).
13646		
13647		
13648	<i>SHELL</i>	Determine the preferred command line interpreter for use as the default value of the shell edit option.
13649		
13650	<i>TERM</i>	Determine the name of the terminal type. If this variable is unset or null, an unspecified default terminal type shall be used.
13651		
13652	ASYNCHRONOUS EVENTS	
13653		The following term is used in this and following sections to specify command and asynchronous event actions:
13654		
13655	<i>complete write</i>	A complete write is a write of the entire contents of the edit buffer to a file of a type other than a terminal device, or the saving of the edit buffer caused by the user executing the <i>ex preserve</i> command. Writing the contents of the edit buffer to a temporary file that will be removed when the editor exits shall not be considered a complete write.
13656		
13657		
13658		
13659		
13660		
13661		The following actions shall be taken upon receipt of signals:
13662	<i>SIGINT</i>	If the standard input is not a terminal device, <i>ex</i> shall not write the file or return to command or text input mode, and shall exit with a non-zero exit status.
13663		
13664		Otherwise, if executing an open or visual text input mode command, <i>ex</i> in receipt of SIGINT shall behave identically to its receipt of the <ESC> character.
13665		
13666		Otherwise:
13667	1.	If executing an <i>ex</i> text input mode command, all input lines that have been completely entered shall be resolved into the edit buffer, and any partially entered line shall be discarded.
13668		
13669		

- 13670 2. If there is a currently executing command, it shall be aborted and a message
13671 displayed. Unless otherwise specified by the **ex** or **vi** command descriptions,
13672 it is unspecified whether any lines modified by the executing command
13673 appear modified, or as they were before being modified by the executing
13674 command, in the buffer.
- 13675 If the currently executing command was a motion command, its associated
13676 command shall be discarded.
- 13677 3. If in open or visual command mode, the terminal shall be alerted.
- 13678 4. The editor shall then return to command mode.
- 13679 **SIGCONT** The screen shall be refreshed if in open or visual mode.
- 13680 **SIGHUP** If the edit buffer has been modified since the last complete write, **ex** shall attempt
13681 to save the edit buffer so that it can be recovered later using the **-r** option or the **ex**
13682 **recover** command. The editor shall not write the file or return to command or text
13683 input mode, and shall terminate with a non-zero exit status.
- 13684 **SIGTERM** Refer to SIGHUP.
- 13685 The action taken for all other signals is unspecified.

13686 **STDOUT**

13687 The standard output shall be used only for writing prompts to the user, for informational
13688 messages, and for writing lines from the file.

13689 **STDERR**

13690 The standard error shall be used only for diagnostic messages.

13691 **OUTPUT FILES**

13692 The output from **ex** shall be text files.

13693 **EXTENDED DESCRIPTION**

13694 Only the **ex** mode of the editor is described in this section. See **vi** for additional editing
13695 capabilities available in **ex**.

13696 When an error occurs, **ex** shall write a message. If the terminal supports a standout mode (such
13697 as inverse video), the message shall be written in standout mode. If the terminal does not
13698 support a standout mode, and the edit option **errorbells** is set, an alert action shall precede the
13699 error message.

13700 By default, **ex** shall start in command mode, which shall be indicated by a **:** prompt; see the
13701 **prompt** command. Text input mode can be entered by the **append**, **insert**, or **change** commands;
13702 it can be exited (and command mode re-entered) by typing a period ('.') alone at the beginning
13703 of a line.

13704 **Initialization in ex and vi**

13705 The following symbols are used in this and following sections to specify locations in the edit
13706 buffer:

13707 *alternate and current pathnames*

13708 Two pathnames, named *current* and *alternate*, are maintained by the editor. Any **ex**
13709 commands that take filenames as arguments shall set them as follows:

- 13710 1. If a *file* argument is specified to the **ex edit**, **ex**, or **recover** commands, or if an **ex tag**
13711 command replaces the contents of the edit buffer.

- 13712 a. If the command replaces the contents of the edit buffer, the current pathname
13713 shall be set to the *file* argument or the file indicated by the tag, and the alternate
13714 pathname shall be set to the previous value of the current pathname.
13715 b. Otherwise, the alternate pathname shall be set to the *file* argument.
13716 2. If a *file* argument is specified to the **ex next** command:
13717 a. If the command replaces the contents of the edit buffer, the current pathname
13718 shall be set to the first *file* argument, and the alternate pathname shall be set to
13719 the previous value of the current pathname.
13720 3. If a *file* argument is specified to the **ex file** command, the current pathname shall be set
13721 to the *file* argument, and the alternate pathname shall be set to the previous value of
13722 the current pathname.
13723 4. If a *file* argument is specified to the **ex read** and **write** commands (that is, when
13724 reading or writing a file, and not to the program named by the **shell** edit option), or a
13725 *file* argument is specified to the **ex exit** command:
13726 a. If the current pathname has no value, the current pathname shall be set to the *file*
13727 argument.
13728 b. Otherwise, the alternate pathname shall be set to the *file* argument.

13729 If the alternate pathname is set to the previous value of the current pathname when the
13730 current pathname had no previous value, then the alternate pathname shall have no value
13731 as a result.

13732 *current line*

13733 The line of the edit buffer referenced by the cursor. Each command description specifies the
13734 current line after the command has been executed, as the *current line* value. When the edit
13735 buffer contains no lines, the current line shall be zero; see **Addressing in ex** (on page 359).

13736 *current column*

13737 The current display line column occupied by the cursor. (The columns shall be numbered
13738 beginning at 1.) Each command description specifies the current column after the command
13739 has been executed, as the *current column* value. This column is an *ideal* column that is
13740 remembered over the lifetime of the editor. The actual display line column upon which the
13741 cursor rests may be different from the current column; see the cursor positioning discussion
13742 in **Command Descriptions in vi** (on page 985).

13743 *set to non-<blank>*

13744 A description for a current column value, meaning that the current column shall be set to
13745 the last display line column on which is displayed any part of the first non-<blank> of the
13746 line. If the line has no non-<blank> non-<newline>s, the current column shall be set to the
13747 last display line column on which is displayed any part of the last non-<newline> in the
13748 line. If the line is empty, the current column shall be set to column position 1.

13749 The length of lines in the edit buffer may be limited to {LINE_MAX} bytes. In open and visual
13750 mode, the length of lines in the edit buffer may be limited to the number of characters that will
13751 fit in the display. If either limit is exceeded during editing, an error message shall be written. If
13752 either limit is exceeded by a line read in from a file, an error message shall be written and the
13753 edit session may be terminated.

13754 If the editor stops running due to any reason other than a user command, and the edit buffer has
13755 been modified since the last complete write, it shall be equivalent to a SIGHUP asynchronous
13756 event. If the system crashes, it shall be equivalent to a SIGHUP asynchronous event.

13757 During initialization (before the first file is copied into the edit buffer or any user commands
13758 from the terminal are processed) the following shall occur:

13759 1. If the environment variable *EXINIT* is set, the editor shall execute the *ex* commands
13760 contained in that variable.

13761 2. If the *EXINIT* variable is not set, and all of the following are true:

13762 a. The *HOME* environment variable is not null and not empty.

13763 b. The file *.exrc* in the directory referred to by the *HOME* environment variable:

13764 1. Exists

13765 2. Is owned by the same user ID as the real user ID of the process or the process
13766 has appropriate privileges

13767 3. Is not writable by anyone other than the owner

13768 the editor shall execute the *ex* commands contained in that file.

13769 3. If and only if all of the following are true:

13770 a. The current directory is not referred to by the *HOME* environment variable.

13771 b. A command in the *EXINIT* environment variable or a command in the *.exrc* file in the
13772 directory referred to by the *HOME* environment variable sets the editor option *exrc*.

13773 c. The *.exrc* file in the current directory:

13774 1. Exists

13775 2. Is owned by the same user ID as the real user ID of the process, or by one of a
13776 set of implementation-defined user IDs

13777 3. Is not writable by anyone other than the owner

13778 the editor shall attempt to execute the *ex* commands contained in that file.

13779 Lines in any *.exrc* file that are blank lines shall be ignored. If any *.exrc* file exists, but is not read
13780 for ownership or permission reasons, it shall be an error.

13781 After the *EXINIT* variable and any *.exrc* files are processed, the first file specified by the user
13782 shall be edited, as follows:

13783 1. If the user specified the *-t* option, the effect shall be as if the *ex tag* command was entered
13784 with the specified argument, with the exception that if tag processing does not result in a
13785 file to edit, the effect shall be as described in step 3. below.

13786 2. Otherwise, if the user specified any command line *file* arguments, the effect shall be as if
13787 the *ex edit* command was entered with the first of those arguments as its *file* argument.

13788 3. Otherwise, the effect shall be as if the *ex edit* command was entered with a nonexistent
13789 filename as its *file* argument. It is unspecified whether this action shall set the current
13790 pathname. In an implementation where this action does not set the current pathname, any
13791 editor command using the current pathname shall fail until an editor command sets the
13792 current pathname.

13793 If the *-r* option was specified, the first time a file in the initial argument list or a file specified by
13794 the *-t* option is edited, if recovery information has previously been saved about it, that
13795 information shall be recovered and the editor shall behave as if the contents of the edit buffer
13796 have already been modified. If there are multiple instances of the file to be recovered, the one
13797 most recently saved shall be recovered, and an informational message that there are previous

13798 versions of the file that can be recovered shall be written. If no recovery information about a file
13799 is available, an informational message to this effect shall be written, and the edit shall proceed as
13800 usual.

13801 If the **-c** option was specified, the first time a file that already exists (including a file that might
13802 not exist but for which recovery information is available, when the **-r** option is specified)
13803 replaces or initializes the contents of the edit buffer, the current line shall be set to the last line of
13804 the edit buffer, the current column shall be set to non-<blank>, and the **ex** commands specified
13805 with the **-c** option shall be executed. In this case, the current line and current column shall not be
13806 set as described for the command associated with the replacement or initialization of the edit
13807 buffer contents. However, if the **-t** option or a **tag** command is associated with this action, the **-c**
13808 option commands shall be executed and then the movement to the tag shall be performed.

13809 The current argument list shall initially be set to the filenames specified by the user on the
13810 command line. If no filenames are specified by the user, the current argument list shall be empty.
13811 If the **-t** option was specified, it is unspecified whether any filename resulting from tag
13812 processing shall be prepended to the current argument list. In the case where the filename is
13813 added as a prefix to the current argument list, the current argument list reference shall be set to
13814 that filename. In the case where the filename is not added as a prefix to the current argument
13815 list, the current argument list reference shall logically be located before the first of the filenames
13816 specified on the command line (for example, a subsequent **ex next** command shall edit the first
13817 filename from the command line). If the **-t** option was not specified, the current argument list
13818 reference shall be to the first of the filenames on the command line.

13819 Addressing in ex

13820 Addressing in **ex** relates to the current line and the current column; the address of a line is its 1-
13821 based line number, the address of a column is its 1-based count from the beginning of the line.
13822 Generally, the current line is the last line affected by a command. The current line number is the
13823 address of the current line. In each command description, the effect of the command on the
13824 current line number and the current column is described.

13825 Addresses are constructed as follows:

- 13826 1. The character '.' (period) shall address the current line.
- 13827 2. The character '\$' shall address the last line of the edit buffer.
- 13828 3. The positive decimal number *n* shall address the *n*th line of the edit buffer.
- 13829 4. The address "'x'" refers to the line marked with the mark name character 'x', which shall
13830 be a lowercase letter from the portable character set or one of the characters ''`' or ''''. It
13831 shall be an error if the line that was marked is not currently present in the edit buffer or the
13832 mark has not been set. Lines can be marked with the **ex mark** or **k** commands, or the **vi m**
13833 command.
- 13834 5. A regular expression enclosed by slashes ('//') shall address the first line found by
13835 searching forwards from the line following the current line toward the end of the edit
13836 buffer and stopping at the first line for which the line excluding the terminating <newline>
13837 matches the regular expression. As stated in **Regular Expressions in ex** (on page 389), an
13838 address consisting of a null regular expression delimited by slashes " // " shall address the
13839 next line for which the line excluding the terminating <newline> matches the last regular
13840 expression encountered. In addition, the second slash can be omitted at the end of a
13841 command line. If the **wrapscan** edit option is set, the search shall wrap around to the
13842 beginning of the edit buffer and continue up to and including the current line, so that the
13843 entire edit buffer is searched. Within the regular expression, the sequence "\/" shall
13844 represent a literal slash instead of the regular expression delimiter.

- 13845 6. A regular expression enclosed in question marks ('?') shall address the first line found by
13846 searching backwards from the line preceding the current line toward the beginning of the
13847 edit buffer and stopping at the first line for which the line excluding the terminating
13848 <newline> matches the regular expression. An address consisting of a null regular
13849 expression delimited by question marks "???" shall address the previous line for which the
13850 line excluding the terminating <newline> matches the last regular expression encountered.
13851 In addition, the second question mark can be omitted at the end of a command line. If the
13852 **wrapscan** edit option is set, the search shall wrap around from the beginning of the edit
13853 buffer to the end of the edit buffer and continue up to and including the current line, so
13854 that the entire edit buffer is searched. Within the regular expression, the sequence "\?"
13855 shall represent a literal question mark instead of the RE delimiter.
- 13856 7. A plus sign ('+') or a minus sign ('-') followed by a decimal number shall address the
13857 current line plus or minus the number. A '+' or '-' not followed by a decimal number
13858 shall address the current line plus or minus 1.

13859 Addresses can be followed by zero or more address offsets, optionally <blank>-separated.
13860 Address offsets are constructed as follows:

- 13861 1. A '+' or '-' immediately followed by a decimal number shall add (subtract) the
13862 indicated number of lines to (from) the address. A '+' or '-' not followed by a decimal
13863 number shall add (subtract) 1 to (from) the address.
- 13864 2. A decimal number shall add the indicated number of lines to the address.

13865 It shall not be an error for an intermediate address value to be less than zero or greater than the
13866 last line in the edit buffer. It shall be an error for the final address value to be less than zero or
13867 greater than the last line in the edit buffer.

13868 Commands take zero, one, or two addresses; see the descriptions of *1addr* and *2addr* in
13869 **Command Descriptions in ex** (on page 366). If more than the required number of addresses are
13870 provided to a command that requires zero addresses, it shall be an error. Otherwise, if more than
13871 the required number of addresses are provided to a command, the addresses specified first shall
13872 be evaluated and then discarded until the maximum number of valid addresses remain.

13873 Addresses shall be separated from each other by a comma (',') or a semicolon (';'). If no
13874 address is specified before or after a comma or semicolon separator, it shall be as if the address
13875 of the current line was specified before or after the separator. In the case of a semicolon
13876 separator, the current line ('.') shall be set to the first address, and only then will the next
13877 address be calculated. This feature can be used to determine the starting line for forwards and
13878 backwards searches (see rules 5. and 6.).

13879 A percent sign ('%') shall be equivalent to entering the two addresses "1, \$".

13880 Any delimiting <blank>s between addresses, address separators, or address offsets shall be
13881 discarded.

13882 **Command Line Parsing in ex**

13883 The following symbol is used in this and following sections to describe parsing behavior:

13884 **escape** If a character is referred to as "backslash-escaped" or "<control>-V-escaped," it
13885 shall mean that the character acquired or lost a special meaning by virtue of being
13886 preceded, respectively, by a backslash or <control>-V character. Unless otherwise
13887 specified, the escaping character shall be discarded at that time and shall not be
13888 further considered for any purpose.

13889 Command-line parsing shall be done in the following steps. For each step, characters already
13890 evaluated shall be ignored; that is, the phrase ‘‘leading character’’ refers to the next character
13891 that has not yet been evaluated.

- 13892 1. Leading colon characters shall be skipped.
 - 13893 2. Leading <blank>s shall be skipped.
 - 13894 3. If the leading character is a double-quote character, the characters up to and including the
13895 next non-backslash-escaped <newline> shall be discarded, and any subsequent characters
13896 shall be parsed as a separate command.
 - 13897 4. Leading characters that can be interpreted as addresses shall be evaluated; see **Addressing**
13898 **in ex** (on page 359).
 - 13899 5. Leading <blank>s shall be skipped.
 - 13900 6. If the next character is a vertical-line character or a <newline>:
 - 13901 a. If the next character is a <newline>:
 - 13902 1. If **ex** is in open or visual mode, the current line shall be set to the last address
13903 specified, if any.
 - 13904 2. Otherwise, if the last command was terminated by a vertical-line character, no
13905 action shall be taken; for example, the command " | |<newline>" shall
13906 execute two implied commands, not three.
 - 13907 3. Otherwise, step 6.b. shall apply.
 - 13908 b. Otherwise, the implied command shall be the **print** command. The last #, p, and l
13909 flags specified to any **ex** command shall be remembered and shall apply to this
13910 implied command. Executing the **ex number, print, or list** command shall set the
13911 remembered flags to #, nothing, and l, respectively, plus any other flags specified for
13912 that execution of the **number, print, or list** command.
 - 13913 If **ex** is not currently performing a **global** or v command, and no address or count is
13914 specified, the current line shall be incremented by 1 before the command is executed.
13915 If incrementing the current line would result in an address past the last line in the
13916 edit buffer, the command shall fail, and the increment shall not happen.
 - 13917 c. The <newline> or vertical-line character shall be discarded and any subsequent
13918 characters shall be parsed as a separate command.
- 13919 7. The command name shall be comprised of the next character (if the character is not
13920 alphabetic), or the next character and any subsequent alphabetic characters (if the
13921 character is alphabetic), with the following exceptions:
- 13922 a. Commands that consist of any prefix of the characters in the command name **delete**,
13923 followed immediately by any of the characters '1', 'p', '+', '−', or '#' shall be
13924 interpreted as a **delete** command, followed by a <blank>, followed by the characters
13925 that were not part of the prefix of the **delete** command. The maximum number of
13926 characters shall be matched to the command name **delete**; for example, "de1" shall
13927 not be treated as "de" followed by the flag l.
 - 13928 b. Commands that consist of the character 'k', followed by a character that can be
13929 used as the name of a mark, shall be equivalent to the mark command followed by a
13930 <blank>, followed by the character that followed the 'k'.
 - 13931 c. Commands that consist of the character 's', followed by characters that could be
13932 interpreted as valid options to the **s** command, shall be the equivalent of the **s**

13933 command, without any pattern or replacement values, followed by a <blank>,
 13934 followed by the characters after the 's'.

- 13935 8. The command name shall be matched against the possible command names, and a
 13936 command name that contains a prefix matching the characters specified by the user shall
 13937 be the executed command. In the case of commands where the characters specified by the
 13938 user could be ambiguous, the executed command shall be as follows:

a	append	n	next	t	t
c	change	p	print	u	undo
ch	change	pr	print	un	undo
e	edit	r	read	v	v
m	move	re	read	w	write
ma	mark	s	s		

13945 Implementation extensions with names causing similar ambiguities shall not be checked
 13946 for a match until all possible matches for commands specified by IEEE Std 1003.1-2001
 13947 have been checked.

- 13948 9. If the command is a ! command, or if the command is a **read** command followed by zero
 13949 or more <blank>s and a !, or if the command is a **write** command followed by one or more
 13950 <blank>s and a !, the rest of the command shall include all characters up to a non-
 13951 backslash-escaped <newline>. The <newline> shall be discarded and any subsequent
 13952 characters shall be parsed as a separate ex command.
- 13953 10. Otherwise, if the command is an **edit**, **ex**, or **next** command, or a **visual** command while in
 13954 open or visual mode, the next part of the command shall be parsed as follows:
- 13955 a. Any '!' character immediately following the command shall be skipped and be part
 13956 of the command.
- 13957 b. Any leading <blank>s shall be skipped and be part of the command.
- 13958 c. If the next character is a '+', characters up to the first non-backslash-escaped
 13959 <newline> or non-backslash-escaped <blank> shall be skipped and be part of the
 13960 command.
- 13961 d. The rest of the command shall be determined by the steps specified in paragraph 12.
- 13962 11. Otherwise, if the command is a **global**, **open**, **s**, or **v** command, the next part of the
 13963 command shall be parsed as follows:
- 13964 a. Any leading <blank>s shall be skipped and be part of the command.
- 13965 b. If the next character is not an alphanumeric, double-quote, <newline>, backslash, or
 13966 vertical-line character:
- 13967 1. The next character shall be used as a command delimiter.
- 13968 2. If the command is a **global**, **open**, or **v** command, characters up to the first
 13969 non-backslash-escaped <newline>, or first non-backslash-escaped delimiter
 13970 character, shall be skipped and be part of the command.
- 13971 3. If the command is an **s** command, characters up to the first non-backslash-
 13972 escaped <newline>, or second non-backslash-escaped delimiter character, shall
 13973 be skipped and be part of the command.
- 13974 c. If the command is a **global** or **v** command, characters up to the first non-backslash-
 13975 escaped <newline> shall be skipped and be part of the command.

- 13976 d. Otherwise, the rest of the command shall be determined by the steps specified in
13977 paragraph 12.
- 13978 12. Otherwise:
- 13979 a. If the command was a **map**, **unmap**, **abbreviate**, or **unabbreviate** command,
13980 characters up to the first non-<control>-V-escaped <newline>, vertical-line, or
13981 double-quote character shall be skipped and be part of the command.
- 13982 b. Otherwise, characters up to the first non-backslash-escaped <newline>, vertical-line,
13983 or double-quote character shall be skipped and be part of the command.
- 13984 c. If the command was an **append**, **change**, or **insert** command, and the step 12.b.
13985 ended at a vertical-line character, any subsequent characters, up to the next non-
13986 backslash-escaped <newline> shall be used as input text to the command.
- 13987 d. If the command was ended by a double-quote character, all subsequent characters,
13988 up to the next non-backslash-escaped <newline>, shall be discarded.
- 13989 e. The terminating <newline> or vertical-line character shall be discarded and any
13990 subsequent characters shall be parsed as a separate ex command.

13991 Command arguments shall be parsed as described by the Synopsis and Description of each
13992 individual ex command. This parsing shall not be <blank>-sensitive, except for the ! argument,
13993 which must follow the command name without intervening <blank>s, and where it would
13994 otherwise be ambiguous. For example, *count* and *flag* arguments need not be <blank>-separated
13995 because "d22p" is not ambiguous, but *file* arguments to the ex **next** command must be
13996 separated by one or more <blank>s. Any <blank> in command arguments for the **abbreviate**,
13997 **unabbreviate**, **map**, and **unmap** commands can be <control>-V-escaped, in which case the
13998 <blank> shall not be used as an argument delimiter. Any <blank> in the command argument for
13999 any other command can be backslash-escaped, in which case that <blank> shall not be used as
14000 an argument delimiter.

14001 Within command arguments for the **abbreviate**, **unabbreviate**, **map**, and **unmap** commands,
14002 any character can be <control>-V-escaped. All such escaped characters shall be treated literally
14003 and shall have no special meaning. Within command arguments for all other ex commands that
14004 are not regular expressions or replacement strings, any character that would otherwise have a
14005 special meaning can be backslash-escaped. Escaped characters shall be treated literally, without
14006 special meaning as shell expansion characters or '!', '%', and '#' expansion characters. See
14007 **Regular Expressions in ex** (on page 389) and **Replacement Strings in ex** (on page 389) for
14008 descriptions of command arguments that are regular expressions or replacement strings.

14009 Non-backslash-escaped '%' characters appearing in *file* arguments to any ex command shall be
14010 replaced by the current pathname; unescaped '#' characters shall be replaced by the alternate
14011 pathname. It shall be an error if '%' or '#' characters appear unescaped in an argument and
14012 their corresponding values are not set.

14013 Non-backslash-escaped '!' characters in the arguments to either the ex ! command or the open
14014 and visual mode ! command, or in the arguments to the ex **read** command, where the first non-
14015 <blank> after the command name is a '!' character, or in the arguments to the ex **write**
14016 command where the command name is followed by one or more <blank>s and the first non-
14017 <blank> after the command name is a '!' character, shall be replaced with the arguments to the
14018 last of those three commands as they appeared after all unescaped '%', '#', and '!' characters
14019 were replaced. It shall be an error if '!' characters appear unescaped in one of these commands
14020 and there has been no previous execution of one of these commands.

14021 If an error occurs during the parsing or execution of an ex command:

- 14022 • An informational message to this effect shall be written. Execution of the **ex** command shall
14023 stop, and the cursor (for example, the current line and column) shall not be further modified.
- 14024 • If the **ex** command resulted from a map expansion, all characters from that map expansion
14025 shall be discarded, except as otherwise specified by the **map** command.
- 14026 • Otherwise, if the **ex** command resulted from the processing of an *EXINIT* environment
14027 variable, a **.exrc** file, a **:source** command, a **-c** option, or a **+command** specified to an **ex edit**,
14028 **ex**, **next**, or **visual** command, no further commands from the source of the commands shall
14029 be executed.
- 14030 • Otherwise, if the **ex** command resulted from the execution of a buffer or a **global** or **v**
14031 command, no further commands caused by the execution of the buffer or the **global** or **v**
14032 command shall be executed.
- 14033 • Otherwise, if the **ex** command was not terminated by a <newline>, all characters up to and
14034 including the next non-backslash-escaped <newline> shall be discarded.

14035 Input Editing in ex

14036 The following symbol is used in this and the following sections to specify command actions:

14037 **word** In the POSIX locale, a word consists of a maximal sequence of letters, digits, and
14038 underscores, delimited at both ends by characters other than letters, digits, or
14039 underscores, or by the beginning or end of a line or the edit buffer.

14040 When accepting input characters from the user, in either **ex** command mode or **ex** text input
14041 mode, **ex** shall enable canonical mode input processing, as defined in the System Interfaces
14042 volume of IEEE Std 1003.1-2001.

14043 If in **ex** text input mode:

1. If the **number** edit option is set, **ex** shall prompt for input using the line number that would
14045 be assigned to the line if it is entered, in the format specified for the **ex number** command.
2. If the **autoindent** edit option is set, **ex** shall prompt for input using **autoindent** characters,
14047 as described by the **autoindent** edit option. **autoindent** characters shall follow the line
14048 number, if any.

14049 If in **ex** command mode:

1. If the **prompt** edit option is set, input shall be prompted for using a single ' : ' character;
14051 otherwise, there shall be no prompt.

14052 The input characters in the following sections shall have the following effects on the input line.

14053 Scroll

14054 **Synopsis:** eof

14055 See the description of the **stty eof** character in **stty**.

14056 If in **ex** command mode:

- 14057 If the **eof** character is the first character entered on the line, the line shall be evaluated as if it
14058 contained two characters: a <control>-D and a <newline>.
14059 Otherwise, the **eof** character shall have no special meaning.

- 14060 If in ex text input mode:
- 14061 If the cursor follows an **autoindent** character, the **autoindent** characters in the line shall be modified so that a part of the next text input character will be displayed on the first column in the line after the previous **shiftwidth** edit option column boundary, and the user shall be prompted again for input for the same line.
- 14065 Otherwise, if the cursor follows a '0', which follows an **autoindent** character, and the '0' was the previous text input character, the '0' and all **autoindent** characters in the line shall be discarded, and the user shall be prompted again for input for the same line.
- 14068 Otherwise, if the cursor follows a '^', which follows an **autoindent** character, and the '^' was the previous text input character, the '^' and all **autoindent** characters in the line shall be discarded, and the user shall be prompted again for input for the same line. In addition, the **autoindent** level for the next input line shall be derived from the same line from which the **autoindent** level for the current input line was derived.
- 14073 Otherwise, if there are no **autoindent** or text input characters in the line, the **eof** character shall be discarded.
- 14075 Otherwise, the **eof** character shall have no special meaning.
- 14076 **<newline>**
- 14077 *Synopsis:* **<newline>**
14078 **<control>-J**
- 14079 If in ex command mode:
- 14080 Cause the command line to be parsed; **<control>-J** shall be mapped to the **<newline>** for this purpose.
- 14082 If in ex text input mode:
- 14083 Terminate the current line. If there are no characters other than **autoindent** characters on the line, all characters on the line shall be discarded.
- 14085 Prompt for text input on a new line after the current line. If the **autoindent** edit option is set, an appropriate number of **autoindent** characters shall be added as a prefix to the line as described by the **ex autoindent** edit option.
- 14088 **<backslash>**
- 14089 *Synopsis:* **<backslash>**
- 14090 Allow the entry of a subsequent **<newline>** or **<control>-J** as a literal character, removing any special meaning that it may have to the editor during text input mode. The backslash character shall be retained and evaluated when the command line is parsed, or retained and included when the input text becomes part of the edit buffer.

14094 <control>-V

14095 *Synopsis:* <control>-V

14096 Allow the entry of any subsequent character as a literal character, removing any special meaning
14097 that it may have to the editor during text input mode. The <control>-V character shall be
14098 discarded before the command line is parsed or the input text becomes part of the edit buffer.

14099 If the “literal next” functionality is performed by the underlying system, it is implementation-
14100 defined whether a character other than <control>-V performs this function.

14101 <control>-W

14102 *Synopsis:* <control>-W

14103 Discard the <control>-W, and the word previous to it in the input line, including any <blank>s
14104 following the word and preceding the <control>-W. If the “word erase” functionality is
14105 performed by the underlying system, it is implementation-defined whether a character other
14106 than <control>-W performs this function.

14107 Command Descriptions in ex

14108 The following symbols are used in this section to represent command modifiers. Some of these
14109 modifiers can be omitted, in which case the specified defaults shall be used.

14110 *addr* A single line address, given in any of the forms described in **Addressing in ex** (on
14111 page 359); the default shall be the current line ('.'), unless otherwise specified.

14112 If the line address is zero, it shall be an error, unless otherwise specified in the
14113 following command descriptions.

14114 If the edit buffer is empty, and the address is specified with a command other than
14115 '=', **append**, **insert**, **open**, **put**, **read**, or **visual**, or the address is not zero, it shall be
14116 an error.

14117 *2addr* Two addresses specifying an inclusive range of lines. If no addresses are specified,
14118 the default for *2addr* shall be the current line only (".."), unless otherwise
14119 specified in the following command descriptions. If one address is specified, *2addr*
14120 shall specify that line only, unless otherwise specified in the following command
14121 descriptions.

14122 It shall be an error if the first address is greater than the second address.

14123 If the edit buffer is empty, and the two addresses are specified with a command
14124 other than the !, **write**, **wq**, or **xit** commands, or either address is not zero, it shall
14125 be an error.

14126 *count* A positive decimal number. If *count* is specified, it shall be equivalent to specifying
14127 an additional address to the command, unless otherwise specified by the following
14128 command descriptions. The additional address shall be equal to the last address
14129 specified to the command (either explicitly or by default) plus *count*-1.

14130 If this would result in an address greater than the last line of the edit buffer, it shall
14131 be corrected to equal the last line of the edit buffer.

14132 *flags* One or more of the characters '+', '-', '#', 'p', or 'l' (ell). The flag characters
14133 can be <blank>-separated, and in any order or combination. The characters '#' ,
14134 'p' , and 'l' shall cause lines to be written in the format specified by the **print**
14135 command with the specified *flags*.

14136

The lines to be written are as follows:

14137

14138

1. All edit buffer lines written during the execution of the **ex &, ^, list, number, open, print, s, visual, and z** commands shall be written as specified by *flags*.

14139

14140

14141

2. After the completion of an **ex** command with a flag as an argument, the current line shall be written as specified by *flags*, unless the current line was the last line written by the command.

14142

14143

14144

14145

The characters '+' and '-' cause the value of the current line after the execution of the **ex** command to be adjusted by the offset address as described in **Addressing in ex** (on page 359). This adjustment shall occur before the current line is written as described in 2. above.

14146

The default for *flags* shall be none.

14147

buffer

14148

14149

14150

14151

14152

14153

14154

One of a number of named areas for holding text. The named buffers are specified by the alphanumeric characters of the POSIX locale. There shall also be one "unnamed" buffer. When no buffer is specified for editor commands that use a buffer, the unnamed buffer shall be used. Commands that store text into buffers shall store the text as it was before the command took effect, and shall store text occurring earlier in the file before text occurring later in the file, regardless of how the text region was specified. Commands that store text into buffers shall store the text into the unnamed buffer as well as any specified buffer.

14155

14156

In **ex** commands, buffer names are specified as the name by itself. In **open** or **visual** mode commands the name is preceded by a double quote (' ") character.

14157

14158

14159

14160

If the specified buffer name is an uppercase character, and the buffer contents are to be modified, the buffer shall be appended to rather than being overwritten. If the buffer is not being modified, specifying the buffer name in lowercase and uppercase shall have identical results.

14161

14162

14163

14164

14165

14166

14167

14168

14169

14170

14171

14172

There shall also be buffers named by the numbers 1 through 9. In **open** and **visual** mode, if a region of text including characters from more than a single line is being modified by the **vi c** or **d** commands, the motion character associated with the **c** or **d** commands specifies that the buffer text shall be in line mode, or the commands %, '^', '?', '(', ')', 'N', 'n', '{', or '}' are used to define a region of text for the **c** or **d** commands, the contents of buffers 1 through 8 shall be moved into the buffer named by the next numerically greater value, the contents of buffer 9 shall be discarded, and the region of text shall be copied into buffer 1. This shall be in addition to copying the text into a user-specified buffer or unnamed buffer, or both. Numeric buffers can be specified as a source buffer for **open** and **visual** mode commands; however, specifying a numeric buffer as the write target of an **open** or **visual** mode command shall have unspecified results.

14173

14174

14175

14176

14177

14178

14179

14180

14181

The text of each buffer shall have the characteristic of being in either line or character mode. Appending text to a non-empty buffer shall set the mode to match the characteristic of the text being appended. Appending text to a buffer shall cause the creation of at least one additional line in the buffer. All text stored into buffers by **ex** commands shall be in line mode. The **ex** commands that use buffers as the source of text specify individually how buffers of different modes are handled. Each **open** or **visual** mode command that uses buffers for any purpose specifies individually the mode of the text stored into the buffer and how buffers of different modes are handled.

14182 *file* Command text used to derive a pathname. The default shall be the current
14183 pathname, as defined previously, in which case, if no current pathname has yet
14184 been established it shall be an error, except where specifically noted in the
14185 individual command descriptions that follow. If the command text contains any of
14186 the characters '~', '{', '[', '*', '?', '\$', '^', ',', '"', and '\', it shall be
14187 subjected to the process of "shell expansions", as described below; if more than a
14188 single pathname results and the command expects only one, it shall be an error.

14189 The process of shell expansions in the editor shall be done as follows. The ex utility
14190 shall pass two arguments to the program named by the shell edit option; the first
14191 shall be -c, and the second shall be the string "echo" and the command text as a
14192 single argument. The standard output and standard error of that command shall
14193 replace the command text.

14194 ! A character that can be appended to the command name to modify its operation,
14195 as detailed in the individual command descriptions. With the exception of the ex
14196 **read**, **write**, and ! commands, the '!' character shall only act as a modifier if there
14197 are no <blank>s between it and the command name.

14198 *remembered search direction*

14199 The vi commands N and n begin searching in a forwards or backwards direction in
14200 the edit buffer based on a remembered search direction, which is initially unset,
14201 and is set by the ex **global**, **v**, **s**, and **tag** commands, and the vi / and ? commands.

14202 **Abbreviate**

14203 *Synopsis:* ab[*abbreviate*][*lhs rhs*]

14204 If *lhs* and *rhs* are not specified, write the current list of abbreviations and do nothing more.

14205 Implementations may restrict the set of characters accepted in *lhs* or *rh*, except that printable
14206 characters and <blank>s shall not be restricted. Additional restrictions shall be implementation-
14207 defined.

14208 In both *lhs* and *rhs*, any character may be escaped with a <control>-V, in which case the
14209 character shall not be used to delimit *lhs* from *rhs*, and the escaping <control>-V shall be
14210 discarded.

14211 In open and visual text input mode, if a non-word or <ESC> character that is not escaped by a
14212 <control>-V character is entered after a word character, a check shall be made for a set of
14213 characters matching *lhs*, in the text input entered during this command. If it is found, the effect
14214 shall be as if *rhs* was entered instead of *lhs*.

14215 The set of characters that are checked is defined as follows:

- 14216 1. If there are no characters inserted before the word and non-word or <ESC> characters that
14217 triggered the check, the set of characters shall consist of the word character.
- 14218 2. If the character inserted before the word and non-word or <ESC> characters that triggered
14219 the check is a word character, the set of characters shall consist of the characters inserted
14220 immediately before the triggering characters that are word characters, plus the triggering
14221 word character.
- 14222 3. If the character inserted before the word and non-word or <ESC> characters that triggered
14223 the check is not a word character, the set of characters shall consist of the characters that
14224 were inserted before the triggering characters that are neither <blank>s nor word
14225 characters, plus the triggering word character.

14226 It is unspecified whether the *lhs* argument entered for the **ex abbreviate** and **unabbreviate**
14227 commands is replaced in this fashion. Regardless of whether or not the replacement occurs, the
14228 effect of the command shall be as if the replacement had not occurred.

14229 *Current line*: Unchanged.

14230 *Current column*: Unchanged.

14231 Append

14232 *Synopsis:* [1addr] a[ppend][!]

14233 Enter **ex** text input mode; the input text shall be placed after the specified line. If line zero is
14234 specified, the text shall be placed at the beginning of the edit buffer.

14235 This command shall be affected by the **number** and **autoindent** edit options; following the
14236 command name with ‘!’ shall cause the **autoindent** edit option setting to be toggled for the
14237 duration of this command only.

14238 *Current line*: Set to the last input line; if no lines were input, set to the specified line, or to the
14239 first line of the edit buffer if a line of zero was specified, or zero if the edit buffer is empty.

14240 *Current column*: Set to non-<blank>.

14241 Arguments

14242 *Synopsis:* ar[gs]

14243 Write the current argument list, with the current argument-list entry, if any, between ‘[’ and
14244 ‘]’ characters.

14245 *Current line*: Unchanged.

14246 *Current column*: Unchanged.

14247 Change

14248 *Synopsis:* [2addr] c[hang]e[!][count]

14249 Enter **ex** text input mode; the input text shall replace the specified lines. The specified lines shall
14250 be copied into the unnamed buffer, which shall become a line mode buffer.

14251 This command shall be affected by the **number** and **autoindent** edit options; following the
14252 command name with ‘!’ shall cause the **autoindent** edit option setting to be toggled for the
14253 duration of this command only.

14254 *Current line*: Set to the last input line; if no lines were input, set to the line before the first
14255 address, or to the first line of the edit buffer if there are no lines preceding the first address, or to
14256 zero if the edit buffer is empty.

14257 *Current column*: Set to non-<blank>.

14258 **Change Directory**

14259 *Synopsis:* `chd[ir][!][directory]`
14260 `cd[!][directory]`

14261 Change the current working directory to *directory*.

14262 If no *directory* argument is specified, and the *HOME* environment variable is set to a non-null
14263 and non-empty value, *directory* shall default to the value named in the *HOME* environment
14264 variable. If the *HOME* environment variable is empty or is undefined, the default value of
14265 *directory* is implementation-defined.

14266 If no '!' is appended to the command name, and the edit buffer has been modified since the
14267 last complete write, and the current pathname does not begin with a '/', it shall be an error.

14268 *Current line:* Unchanged.

14269 *Current column:* Unchanged.

14270 **Copy**

14271 *Synopsis:* `[2addr] co[py] 1addr [flags]`
14272 `[2addr] t 1addr [flags]`

14273 Copy the specified lines after the specified destination line; line zero specifies that the lines shall
14274 be placed at the beginning of the edit buffer.

14275 *Current line:* Set to the last line copied.

14276 *Current column:* Set to non-<blank>.

14277 **Delete**

14278 *Synopsis:* `[2addr] d[elete][buffer][count][flags]`

14279 Delete the specified lines into a buffer (defaulting to the unnamed buffer), which shall become a
14280 line-mode buffer.

14281 Flags can immediately follow the command name; see **Command Line Parsing in ex** (on page
14282 360).

14283 *Current line:* Set to the line following the deleted lines, or to the last line in the edit buffer if that
14284 line is past the end of the edit buffer, or to zero if the edit buffer is empty.

14285 *Current column:* Set to non-<blank>.

14286 **Edit**

14287 *Synopsis:* `e[dit][!][+command][file]`
14288 `ex[!][+command][file]`

14289 If no '!' is appended to the command name, and the edit buffer has been modified since the
14290 last complete write, it shall be an error.

14291 If *file* is specified, replace the current contents of the edit buffer with the current contents of *file*,
14292 and set the current pathname to *file*. If *file* is not specified, replace the current contents of the
14293 edit buffer with the current contents of the file named by the current pathname. If for any reason
14294 the current contents of the file cannot be accessed, the edit buffer shall be empty.

14295 The *+command* option shall be <blank>-delimited; <blank>s within *+command* can be escaped by
14296 preceding them with a backslash character. The *+command* shall be interpreted as an *ex*
14297 command immediately after the contents of the edit buffer have been replaced and the current

14298 line and column have been set.

14299 If the edit buffer is empty:

14300 *Current line*: Set to 0.

14301 *Current column*: Set to 1.

14302 Otherwise, if executed while in ex command mode or if the *+command* argument is specified:

14303 *Current line*: Set to the last line of the edit buffer.

14304 *Current column*: Set to non-<blank>.

14305 Otherwise, if *file* is omitted or results in the current pathname:

14306 *Current line*: Set to the first line of the edit buffer.

14307 *Current column*: Set to non-<blank>.

14308 Otherwise, if *file* is the same as the last file edited, the line and column shall be set as follows; if
14309 the file was previously edited, the line and column may be set as follows:

14310 *Current line*: Set to the last value held when that file was last edited. If this value is not a valid
14311 line in the new edit buffer, set to the first line of the edit buffer.

14312 *Current column*: If the current line was set to the last value held when the file was last edited, set
14313 to the last value held when the file was last edited. Otherwise, or if the last value is not a valid
14314 column in the new edit buffer, set to non-<blank>.

14315 Otherwise:

14316 *Current line*: Set to the first line of the edit buffer.

14317 *Current column*: Set to non-<blank>.

14318 **File**

14319 *Synopsis:* f[file][file]

14320 If a *file* argument is specified, the alternate pathname shall be set to the current pathname, and
14321 the current pathname shall be set to *file*.

14322 Write an informational message. If the file has a current pathname, it shall be included in this
14323 message; otherwise, the message shall indicate that there is no current pathname. If the edit
14324 buffer contains lines, the current line number and the number of lines in the edit buffer shall be
14325 included in this message; otherwise, the message shall indicate that the edit buffer is empty. If
14326 the edit buffer has been modified since the last complete write, this fact shall be included in this
14327 message. If the **readonly** edit option is set, this fact shall be included in this message. The
14328 message may contain other unspecified information.

14329 *Current line*: Unchanged.

14330 *Current column*: Unchanged.

14331 **Global**

14332 Synopsis: [2addr] g[lobal] /pattern/ [commands]
14333 [2addr] v /pattern/ [commands]

14334 The optional '!' character after the **global** command shall be the same as executing the **v** command.

14336 If *pattern* is empty (for example, " // ") or not specified, the last regular expression used in the
14337 editor command shall be used as the *pattern*. The *pattern* can be delimited by slashes (shown in
14338 the Synopsis), as well as any non-alphanumeric or non-<blank> other than backslash, vertical
14339 line, double quote, or <newline>.

14340 If no lines are specified, the lines shall default to the entire file.

14341 The **global** and **v** commands are logically two-pass operations. First, mark the lines within the
14342 specified lines for which the line excluding the terminating <newline> matches (**global**) or does
14343 not match (**v** or **global!**) the specified pattern. Second, execute the **ex** commands given by
14344 *commands*, with the current line ('.') set to each marked line. If an error occurs during this
14345 process, or the contents of the edit buffer are replaced (for example, by the **ex :edit** command) an
14346 error message shall be written and no more commands resulting from the execution of this
14347 command shall be processed.

14348 Multiple **ex** commands can be specified by entering multiple commands on a single line using a
14349 vertical line to delimit them, or one per line, by escaping each <newline> with a backslash.

14350 If no commands are specified:

- 14351 1. If in **ex** command mode, it shall be as if the **print** command were specified.
- 14352 2. Otherwise, no command shall be executed.

14353 For the **append**, **change**, and **insert** commands, the input text shall be included as part of the
14354 command, and the terminating period can be omitted if the command ends the list of
14355 commands. The **open** and **visual** commands can be specified as one of the commands, in which
14356 case each marked line shall cause the editor to enter open or visual mode. If open or visual mode
14357 is exited using the **vi Q** command, the current line shall be set to the next marked line, and open
14358 or visual mode reentered, until the list of marked lines is exhausted.

14359 The **global**, **v**, and **undo** commands cannot be used in *commands*. Marked lines may be deleted
14360 by commands executed for lines occurring earlier in the file than the marked lines. In this case,
14361 no commands shall be executed for the deleted lines.

14362 If the remembered search direction is not set, the **global** and **v** commands shall set it to forward.

14363 The **autoprint** and **autoindent** edit options shall be inhibited for the duration of the **g** or **v**
14364 command.

14365 *Current line*: If no commands executed, set to the last marked line. Otherwise, as specified for
14366 the executed **ex** commands.

14367 *Current column*: If no commands are executed, set to non-<blank>; otherwise, as specified for the
14368 individual **ex** commands.

14369 **Insert**

14370 *Synopsis:* [*1addr*] *i[nsert][!]*

14371 Enter ex text input mode; the input text shall be placed before the specified line. If the line is zero
14372 or 1, the text shall be placed at the beginning of the edit buffer.

14373 This command shall be affected by the **number** and **autoindent** edit options; following the
14374 command name with '!' shall cause the **autoindent** edit option setting to be toggled for the
14375 duration of this command only.

14376 *Current line:* Set to the last input line; if no lines were input, set to the line before the specified
14377 line, or to the first line of the edit buffer if there are no lines preceding the specified line, or zero
14378 if the edit buffer is empty.

14379 *Current column:* Set to non-<blank>.

14380 **Join**

14381 *Synopsis:* [*2addr*] *j[oin][!][count][flags]*

14382 If *count* is specified:

14383 If no address was specified, the **join** command shall behave as if *2addr* were the current line
14384 and the current line plus *count* (., . + *count*).

14385 If one address was specified, the **join** command shall behave as if *2addr* were the specified
14386 address and the specified address plus *count* (*addr,addr+count*).

14387 If two addresses were specified, the **join** command shall behave as if an additional address,
14388 equal to the last address plus *count* - 1 (*addr1,addr2,addr2+count-1*), was specified.

14389 If this would result in a second address greater than the last line of the edit buffer, it shall be
14390 corrected to be equal to the last line of the edit buffer.

14391 If no *count* is specified:

14392 If no address was specified, the **join** command shall behave as if *2addr* were the current line
14393 and the next line (., . + 1).

14394 If one address was specified, the **join** command shall behave as if *2addr* were the specified
14395 address and the next line (*addr,addr+1*).

14396 Join the text from the specified lines together into a single line, which shall replace the specified
14397 lines.

14398 If a '!' character is appended to the command name, the **join** shall be without modification of
14399 any line, independent of the current locale.

14400 Otherwise, in the POSIX locale, set the current line to the first of the specified lines, and then, for
14401 each subsequent line, proceed as follows:

- 14402 1. Discard leading <space>s from the line to be joined.
- 14403 2. If the line to be joined is now empty, delete it, and skip steps 3 through 5.
- 14404 3. If the current line ends in a <blank>, or the first character of the line to be joined is a ')' character, join the lines without further modification.
- 14405 4. If the last character of the current line is a '.', join the lines with two <space>s between them.

14408 5. Otherwise, join the lines with a single <space> between them.

14409 *Current line*: Set to the first line specified.

14410 *Current column*: Set to non-<blank>.

14411 **List**

14412 *Synopsis*: [2addr] l[ist][count][flags]

14413 This command shall be equivalent to the **ex** command:

14414 [2addr] p[rint][count] l[flags]

14415 See **Print** (on page 378).

14416 **Map**

14417 *Synopsis*: map[!][lhs rhs]

14418 If *lhs* and *rhs* are not specified:

14419 1. If '!' is specified, write the current list of text input mode maps.

14420 2. Otherwise, write the current list of command mode maps.

14421 3. Do nothing more.

14422 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. In both *lhs* and *rhs*, any character can be escaped with a <control>-V, in which case the character shall not be used to delimit *lhs* from *rhs*, and the escaping <control>-V shall be discarded.

14427 If the character '!' is appended to the **map** command name, the mapping shall be effective during open or visual text input mode rather than **open** or **visual** command mode. This allows *lhs* to have two different **map** definitions at the same time: one for command mode and one for text input mode.

14431 For command mode mappings:

14432 When the *lhs* is entered as any part of a **vi** command in open or visual mode (but not as part 14433 of the arguments to the command), the action shall be as if the corresponding *rhs* had been 14434 entered.

14435 If any character in the command, other than the first, is escaped using a <control>-V 14436 character, that character shall not be part of a match to an *lhs*.

14437 It is unspecified whether implementations shall support **map** commands where the *lhs* is 14438 more than a single character in length, where the first character of the *lhs* is printable.

14439 If *lhs* contains more than one character and the first character is '#', followed by a sequence 14440 of digits corresponding to a numbered function key, then when this function key is typed it 14441 shall be mapped to *rhs*. Characters other than digits following a '#' character also represent 14442 the function key named by the characters in the *lhs* following the '#' and may be mapped to 14443 *rhs*. It is unspecified how function keys are named or what function keys are supported.

14444 For text input mode mappings:

14445 When the *lhs* is entered as any part of text entered in open or visual text input modes, the
14446 action shall be as if the corresponding *rhs* had been entered.

14447 If any character in the input text is escaped using a <control>-V character, that character shall
14448 not be part of a match to an *lhs*.

14449 It is unspecified whether the *lhs* text entered for subsequent **map** or **unmap** commands is
14450 replaced with the *rhs* text for the purposes of the screen display; regardless of whether or not
14451 the display appears as if the corresponding *rhs* text was entered, the effect of the command
14452 shall be as if the *lhs* text was entered.

14453 If only part of the *lhs* is entered, it is unspecified how long the editor will wait for additional,
14454 possibly matching characters before treating the already entered characters as not matching the
14455 *lhs*.

14456 The *rhs* characters shall themselves be subject to remapping, unless otherwise specified by the
14457 **remap** edit option, except that if the characters in *lhs* occur as prefix characters in *rhs*, those
14458 characters shall not be remapped.

14459 On block-mode terminals, the mapping need not occur immediately (for example, it may occur
14460 after the terminal transmits a group of characters to the system), but it shall achieve the same
14461 results as if it occurred immediately.

14462 *Current line*: Unchanged.

14463 *Current column*: Unchanged.

14464 **Mark**

14465 *Synopsis:* [1addr] ma[rk] character
14466 [1addr] k character

14467 Implementations shall support *character* values of a single lowercase letter of the POSIX locale
14468 and the characters ' ' and ' ' ; support of other characters is implementation-defined.

14469 If executing the **vi m** command, set the specified mark to the current line and 1-based numbered
14470 character referenced by the current column, if any; otherwise, column position 1.

14471 Otherwise, set the specified mark to the specified line and 1-based numbered first non-<blank>
14472 non-<newline> in the line, if any; otherwise, the last non-<newline> in the line, if any; otherwise,
14473 column position 1.

14474 The mark shall remain associated with the line until the mark is reset or the line is deleted. If a
14475 deleted line is restored by a subsequent **undo** command, any marks previously associated with
14476 the line, which have not been reset, shall be restored as well. Any use of a mark not associated
14477 with a current line in the edit buffer shall be an error.

14478 The marks ' and ' shall be set as described previously, immediately before the following events
14479 occur in the editor:

- 14480 1. The use of '\$' as an **ex** address
- 14481 2. The use of a positive decimal number as an **ex** address
- 14482 3. The use of a search command as an **ex** address
- 14483 4. The use of a mark reference as an **ex** address
- 14484 5. The use of the following open and visual mode commands: <control>-], %, (,), [,], {, }
- 14485 6. The use of the following open and visual mode commands: ', **G**, **H**, **L**, **M**, **z** if the current
14486 line will change as a result of the command

14487 7. The use of the open and visual mode commands: **/**, **?**, **N**, **,** **n** if the current line or column
14488 will change as a result of the command

14489 8. The use of the ex mode commands: **z**, **undo**, **global**, **v**

14490 For rules 1., 2., 3., and 4., the ‘ and ’ marks shall not be set if the **ex** command is parsed as
14491 specified by rule 6.a. in **Command Line Parsing in ex** (on page 360).

14492 For rules 5., 6., and 7., the ‘ and ’ marks shall not be set if the commands are used as motion
14493 commands in open and visual mode.

14494 For rules 1., 2., 3., 4., 5., 6., 7., and 8., the ‘ and ’ marks shall not be set if the command fails.

14495 The ‘ and ’ marks shall be set as described previously, each time the contents of the edit buffer
14496 are replaced (including the editing of the initial buffer), if in open or visual mode, or if in **ex**
14497 mode and the edit buffer is not empty, before any commands or movements (including
14498 commands or movements specified by the **-c** or **-t** options or the **+command** argument) are
14499 executed on the edit buffer. If in open or visual mode, the marks shall be set as if executing the **vi**
14500 **m** command; otherwise, as if executing the **ex mark** command.

14501 When changing from **ex** mode to open or visual mode, if the ‘ and ’ marks are not already set,
14502 the ‘ and ’ marks shall be set as described previously.

14503 *Current line*: Unchanged.

14504 *Current column*: Unchanged.

14505 Move

14506 *Synopsis:* `[2addr] m[ove] 1addr [flags]`

14507 Move the specified lines after the specified destination line. A destination of line zero specifies
14508 that the lines shall be placed at the beginning of the edit buffer. It shall be an error if the
14509 destination line is within the range of lines to be moved.

14510 *Current line*: Set to the last of the moved lines.

14511 *Current column*: Set to non-<blank>.

14512 Next

14513 *Synopsis:* `n[ext][!][+command][file ...]`

14514 If no ‘ ! ’ is appended to the command name, and the edit buffer has been modified since the
14515 last complete write, it shall be an error, unless the file is successfully written as specified by the
14516 **autowrite** option.

14517 If one or more files is specified:

- 14518 1. Set the argument list to the specified filenames.
- 14519 2. Set the current argument list reference to be the first entry in the argument list.
- 14520 3. Set the current pathname to the first filename specified.

14521 Otherwise:

- 14522 1. It shall be an error if there are no more filenames in the argument list after the filename
14523 currently referenced.
- 14524 2. Set the current pathname and the current argument list reference to the filename after the
14525 filename currently referenced in the argument list.

14526 Replace the contents of the edit buffer with the contents of the file named by the current
14527 pathname. If for any reason the contents of the file cannot be accessed, the edit buffer shall be
14528 empty.

14529 This command shall be affected by the **autowrite** and **writeany** edit options.

14530 The **+command** option shall be <blank>-delimited; <blank>s can be escaped by preceding them
14531 with a backslash character. The **+command** shall be interpreted as an **ex** command immediately
14532 after the contents of the edit buffer have been replaced and the current line and column have
14533 been set.

14534 *Current line*: Set as described for the **edit** command.

14535 *Current column*: Set as described for the **edit** command.

14536 Number

14537 *Synopsis:* **[2addr] nu[mber][count][flags]**
14538 **[2addr] #[count][flags]**

14539 These commands shall be equivalent to the **ex** command:

14540 **[2addr] p[rint][count] #[flags]**

14541 See **Print** (on page 378).

14542 Open

14543 *Synopsis:* **[1addr] o[pen] /pattern/ [flags]**

14544 This command need not be supported on block-mode terminals or terminals with insufficient
14545 capabilities. If standard input, standard output, or standard error are not terminal devices, the
14546 results are unspecified.

14547 Enter open mode.

14548 The trailing delimiter can be omitted from *pattern* at the end of the command line. If *pattern* is
14549 empty (for example, " / ") or not specified, the last regular expression used in the editor shall be
14550 used as the pattern. The pattern can be delimited by slashes (shown in the Synopsis), as well as
14551 any alphanumeric, or non-<blank> other than backslash, vertical line, double quote, or
14552 <newline>.

14553 *Current line*: Set to the specified line.

14554 *Current column*: Set to non-<blank>.

14555 Preserve

14556 *Synopsis:* **pre[serve]**

14557 Save the edit buffer in a form that can later be recovered by using the **-r** option or by using the **ex**
14558 **recover** command. After the file has been preserved, a mail message shall be sent to the user.
14559 This message shall be readable by invoking the **mailx** utility. The message shall contain the name
14560 of the file, the time of preservation, and an **ex** command that could be used to recover the file.
14561 Additional information may be included in the mail message.

14562 *Current line*: Unchanged.

14563 *Current column*: Unchanged.

14564 **Print**

14565 *Synopsis:* [*2addr*] *p[rint][count][flags]*

14566 Write the addressed lines. The behavior is unspecified if the number of columns on the display is
14567 less than the number of columns required to write any single character in the lines being written.

14568 Non-printable characters, except for the <tab>, shall be written as implementation-defined
14569 multi-character sequences.

14570 If the # flag is specified or the **number** edit option is set, each line shall be preceded by its line
14571 number in the following format:

14572 "%"dΔΔ", <line number>

14573 If the **I** flag is specified or the **list** edit option is set:

14574 1. The characters listed in the Base Definitions volume of IEEE Std 1003.1-2001, Table 5-1,
14575 Escape Sequences and Associated Actions shall be written as the corresponding escape
14576 sequence.

14577 2. Non-printable characters not in the Base Definitions volume of IEEE Std 1003.1-2001, Table
14578 5-1, Escape Sequences and Associated Actions shall be written as one three-digit octal
14579 number (with a preceding backslash) for each byte in the character (most significant byte
14580 first). If the size of a byte on the system is greater than 9 bits, the format used for non-
14581 printable characters is implementation-defined.

14582 3. The end of each line shall be marked with a '\$', and literal '\$' characters within the line
14583 shall be written with a preceding backslash.

14584 Long lines shall be folded; the length at which folding occurs is unspecified, but should be
14585 appropriate for the output terminal, considering the number of columns of the terminal.

14586 If a line is folded, and the **I** flag is not specified and the **list** edit option is not set, it is unspecified
14587 whether a multi-column character at the folding position is separated; it shall not be discarded.

14588 *Current line:* Set to the last written line.

14589 *Current column:* Unchanged if the current line is unchanged; otherwise, set to non-<blank>.

14590 **Put**

14591 *Synopsis:* [*laddr*] *pu[t][buffer]*

14592 Append text from the specified buffer (by default, the unnamed buffer) to the specified line; line
14593 zero specifies that the text shall be placed at the beginning of the edit buffer. Each portion of a
14594 line in the buffer shall become a new line in the edit buffer, regardless of the mode of the buffer.

14595 *Current line:* Set to the last line entered into the edit buffer.

14596 *Current column:* Set to non-<blank>.

14597 **Quit**

14598 *Synopsis:* *q[uit][!]*

14599 If no '!' is appended to the command name:

1. If the edit buffer has been modified since the last complete write, it shall be an error.
2. If there are filenames in the argument list after the filename currently referenced, and the
last command was not a **quit**, **wq**, **xit**, or **ZZ** (see **Exit** (on page 1019)) command, it shall be
an error.

14604 Otherwise, terminate the editing session.

14605 **Read**

14606 *Synopsis:* [*laddr*] r[ead][!][*file*]

14607 If '!' is not the first non-<blank> to follow the command name, a copy of the specified file shall
14608 be appended into the edit buffer after the specified line; line zero specifies that the copy shall be
14609 placed at the beginning of the edit buffer. The number of lines and bytes read shall be written. If
14610 no *file* is named, the current pathname shall be the default. If there is no current pathname, then
14611 *file* shall become the current pathname. If there is no current pathname or *file* operand, it shall be
14612 an error. Specifying a *file* that is not of type regular shall have unspecified results.

14613 Otherwise, if *file* is preceded by '!', the rest of the line after the '!' shall have '%', '#', and
14614 '!' characters expanded as described in **Command Line Parsing in ex** (on page 360).

14615 The **ex** utility shall then pass two arguments to the program named by the shell edit option; the
14616 first shall be -c and the second shall be the expanded arguments to the **read** command as a
14617 single argument. The standard input of the program shall be set to the standard input of the **ex**
14618 program when it was invoked. The standard error and standard output of the program shall be
14619 appended into the edit buffer after the specified line.

14620 Each line in the copied file or program output (as delimited by <newline>s or the end of the file
14621 or output if it is not immediately preceded by a <newline>), shall be a separate line in the edit
14622 buffer. Any occurrences of <carriage-return> and <newline> pairs in the output shall be treated
14623 as single <newline>s.

14624 The special meaning of the '!' following the **read** command can be overridden by escaping it
14625 with a backslash character.

14626 *Current line:* If no lines are added to the edit buffer, unchanged. Otherwise, if in open or visual
14627 mode, set to the first line entered into the edit buffer. Otherwise, set to the last line entered into
14628 the edit buffer.

14629 *Current column:* Set to non-<blank>.

14630 **Recover**

14631 *Synopsis:* rec[over][!][*file*]

14632 If no '!' is appended to the command name, and the edit buffer has been modified since the
14633 last complete write, it shall be an error.

14634 If no *file* operand is specified, then the current pathname shall be used. If there is no current
14635 pathname or *file* operand, it shall be an error.

14636 If no recovery information has previously been saved about *file*, the **recover** command shall
14637 behave identically to the **edit** command, and an informational message to this effect shall be
14638 written.

14639 Otherwise, set the current pathname to *file*, and replace the current contents of the edit buffer
14640 with the recovered contents of *file*. If there are multiple instances of the file to be recovered, the
14641 one most recently saved shall be recovered, and an informational message that there are
14642 previous versions of the file that can be recovered shall be written. The editor shall behave as if
14643 the contents of the edit buffer have already been modified.

14644 *Current file:* Set as described for the **edit** command.

14645 *Current column:* Set as described for the **edit** command.

14646 **Rewind**

14647 *Synopsis:* `rew[ind][!]`

14648 If no '!' is appended to the command name, and the edit buffer has been modified since the
14649 last complete write, it shall be an error, unless the file is successfully written as specified by the
14650 **autowrite** option.

14651 If the argument list is empty, it shall be an error.

14652 The current argument list reference and the current pathname shall be set to the first filename in
14653 the argument list.

14654 Replace the contents of the edit buffer with the contents of the file named by the current
14655 pathname. If for any reason the contents of the file cannot be accessed, the edit buffer shall be
14656 empty.

14657 This command shall be affected by the **autowrite** and **writeany** edit options.

14658 *Current line:* Set as described for the **edit** command.

14659 *Current column:* Set as described for the **edit** command.

14660 **Set**

14661 *Synopsis:* `set[t][option[=value]] ...[nooption ...][option? ...][all]`

14662 When no arguments are specified, write the value of the **term** edit option and those options
14663 whose values have been changed from the default settings; when the argument *all* is specified,
14664 write all of the option values.

14665 Giving an option name followed by the character '?' shall cause the current value of that
14666 option to be written. The '?' can be separated from the option name by zero or more <blank>s.
14667 The '?' shall be necessary only for Boolean valued options. Boolean options can be given values
14668 by the form **set** *option* to turn them on or **set** **nooption** to turn them off; string and numeric
14669 options can be assigned by the form **set** *option*=*value*. Any <blank>s in strings can be included
14670 as is by preceding each <blank> with an escaping backslash. More than one option can be set or
14671 listed by a single **set** command by specifying multiple arguments, each separated from the next
14672 by one or more <blank>s.

14673 See **Edit Options in ex** (on page 390) for details about specific options.

14674 *Current line:* Unchanged.

14675 *Current column:* Unchanged.

14676 **Shell**

14677 *Synopsis:* `sh[ell]`

14678 Invoke the program named in the **shell** edit option with the single argument **-i** (interactive
14679 mode). Editing shall be resumed when the program exits.

14680 *Current line:* Unchanged.

14681 *Current column:* Unchanged.

14682	Source
14683	<i>Synopsis:</i> <code>so[urce] file</code>
14684	Read and execute <i>ex</i> commands from <i>file</i> . Lines in the file that are blank lines shall be ignored.
14685	<i>Current line:</i> As specified for the individual <i>ex</i> commands.
14686	<i>Current column:</i> As specified for the individual <i>ex</i> commands.
14687	Substitute
14688	<i>Synopsis:</i> <code>[2addr] s[ubstitute][/pattern/repl/[options][count][flags]]</code>
14689	<code>[2addr] &[options][count][flags]</code>
14690	<code>[2addr] ~[options][count][flags]</code>
14691	Replace the first instance of the pattern <i>pattern</i> by the string <i>repl</i> on each specified line. (See Regular Expressions in ex (on page 389) and Replacement Strings in ex (on page 389).) Any non-alphabetic, non-<blank> delimiter other than '\', ' ', double quote, or <newline> can be used instead of '/'. Backslash characters can be used to escape delimiters, backslash characters, and other special characters.
14696	The trailing delimiter can be omitted from <i>pattern</i> or from <i>repl</i> at the end of the command line. If both <i>pattern</i> and <i>repl</i> are not specified or are empty (for example, " // "), the last s command shall be repeated. If only <i>pattern</i> is not specified or is empty, the last regular expression used in the editor shall be used as the pattern. If only <i>repl</i> is not specified or is empty, the pattern shall be replaced by nothing. If the entire replacement pattern is '%', the last replacement pattern to an s command shall be used.
14702	Entering a <carriage-return> in <i>repl</i> (which requires an escaping backslash in <i>ex</i> mode and an escaping <control>-V in open or <i>vi</i> mode) shall split the line at that point, creating a new line in the edit buffer. The <carriage-return> shall be discarded.
14705	If <i>options</i> includes the letter 'g' (global), all non-overlapping instances of the pattern in the line shall be replaced.
14707	If <i>options</i> includes the letter 'c' (confirm), then before each substitution the line shall be written; the written line shall reflect all previous substitutions. On the following line, <space>s shall be written beneath the characters from the line that are before the <i>pattern</i> to be replaced, and '^' characters written beneath the characters included in the <i>pattern</i> to be replaced. The <i>ex</i> utility shall then wait for a response from the user. An affirmative response shall cause the substitution to be done, while any other input shall not make the substitution. An affirmative response shall consist of a line with the affirmative response (as defined by the current locale) at the beginning of the line. This line shall be subject to editing in the same way as the <i>ex</i> command line.
14715	If interrupted (see the ASYNCHRONOUS EVENTS section), any modifications confirmed by the user shall be preserved in the edit buffer after the interrupt.
14717	If the remembered search direction is not set, the s command shall set it to forward.
14718	In the second Synopsis, the & command shall repeat the previous substitution, as if the & command were replaced by:
14720	<code>s/pattern/repl/</code>
14721	where <i>pattern</i> and <i>repl</i> are as specified in the previous s , & , or ~ command.
14722	In the third Synopsis, the ~ command shall repeat the previous substitution, as if the '~~' were replaced by:

14724 *s/pattern/repl/*

14725 where *pattern* shall be the last regular expression specified to the editor, and *repl* shall be from
14726 the previous substitution (including & and ~) command.

14727 These commands shall be affected by the *LC_MESSAGES* environment variable.

14728 *Current line*: Set to the last line in which a substitution occurred, or, unchanged if no
14729 substitution occurred.

14730 *Current column*: Set to non-<blank>.

14731 **Suspend**

14732 *Synopsis:* *su[spend][!]*
14733 *st[op][!]*

14734 Allow control to return to the invoking process; *ex* shall suspend itself as if it had received the
14735 SIGTSTP signal. The suspension shall occur only if job control is enabled in the invoking shell
14736 (see the description of *set -m*).

14737 These commands shall be affected by the **autowrite** and **writeany** edit options.

14738 The current **susp** character (see *stty*) shall be equivalent to the **suspend** command.

14739 **Tag**

14740 *Synopsis:* *ta[g][!]* *tagstring*

14741 The results are unspecified if the format of a tags file is not as specified by the *ctags* utility (see
14742 *ctags*) description.

14743 The **tag** command shall search for *tagstring* in the tag files referred to by the **tag** edit option, in
14744 the order they are specified, until a reference to *tagstring* is found. Files shall be searched from
14745 beginning to end. If no reference is found, it shall be an error and an error message to this effect
14746 shall be written. If the reference is not found, or if an error occurs while processing a file referred
14747 to in the **tag** edit option, it shall be an error, and an error message shall be written at the first
14748 occurrence of such an error.

14749 Otherwise, if the tags file contained a pattern, the pattern shall be treated as a regular expression
14750 used in the editor; for example, for the purposes of the **s** command.

14751 If the *tagstring* is in a file with a different name than the current pathname, set the current
14752 pathname to the name of that file, and replace the contents of the edit buffer with the contents of
14753 that file. In this case, if no '!' is appended to the command name, and the edit buffer has been
14754 modified since the last complete write, it shall be an error, unless the file is successfully written
14755 as specified by the **autowrite** option.

14756 This command shall be affected by the **autowrite**, **tag**, **taglength**, and **writeany** edit options.

14757 *Current line*: If the tags file contained a line number, set to that line number. If the line number is
14758 larger than the last line in the edit buffer, an error message shall be written and the current line
14759 shall be set as specified for the **edit** command.

14760 If the tags file contained a pattern, set to the first occurrence of the pattern in the file. If no
14761 matching pattern is found, an error message shall be written and the current line shall be set as
14762 specified for the **edit** command.

14763 *Current column*: If the tags file contained a line-number reference and that line-number was not
14764 larger than the last line in the edit buffer, or if the tags file contained a pattern and that pattern
14765 was found, set to non-<blank>. Otherwise, set as specified for the **edit** command.

14766 **Unabbreviate**

14767 *Synopsis:* una[bbrev] *lhs*

14768 If *lhs* is not an entry in the current list of abbreviations (see **Abbreviate** (on page 368)), it shall be
14769 an error. Otherwise, delete *lhs* from the list of abbreviations.

14770 *Current line:* Unchanged.

14771 *Current column:* Unchanged.

14772 **Undo**

14773 *Synopsis:* u[ndo]

14774 Reverse the changes made by the last command that modified the contents of the edit buffer,
14775 including **undo**. For this purpose, the **global**, **v**, **open**, and **visual** commands, and commands
14776 resulting from buffer executions and mapped character expansions, are considered single
14777 commands.

14778 If no action that can be undone preceded the **undo** command, it shall be an error.

14779 If the **undo** command restores lines that were marked, the mark shall also be restored unless it
14780 was reset subsequent to the deletion of the lines.

14781 *Current line:*

- 14782 1. If lines are added or changed in the file, set to the first line added or changed.
- 14783 2. Set to the line before the first line deleted, if it exists.
- 14784 3. Set to 1 if the edit buffer is not empty.
- 14785 4. Set to zero.

14786 *Current column:* Set to non-<blank>.

14787 **Unmap**

14788 *Synopsis:* unm[ap][!] *lhs*

14789 If '!' is appended to the command name, and if *lhs* is not an entry in the list of text input mode
14790 map definitions, it shall be an error. Otherwise, delete *lhs* from the list of text input mode map
14791 definitions.

14792 If no '!' is appended to the command name, and if *lhs* is not an entry in the list of command
14793 mode map definitions, it shall be an error. Otherwise, delete *lhs* from the list of command mode
14794 map definitions.

14795 *Current line:* Unchanged.

14796 *Current column:* Unchanged.

14797 **Version**

14798 *Synopsis:* ve[rSION]

14799 Write a message containing version information for the editor. The format of the message is
14800 unspecified.

14801 *Current line:* Unchanged.

14802 *Current column:* Unchanged.

14803 **Visual**

14804 *Synopsis:* [1addr] vi[ual][type][count][flags]

14805 If **ex** is currently in open or visual mode, the Synopsis and behavior of the visual command shall
14806 be the same as the **edit** command, as specified by **Edit** (on page 370).

14807 Otherwise, this command need not be supported on block-mode terminals or terminals with
14808 insufficient capabilities. If standard input, standard output, or standard error are not terminal
14809 devices, the results are unspecified.

14810 If *count* is specified, the value of the **window** edit option shall be set to *count* (as described in
14811 **window** (on page 396)). If the '^' type character was also specified, the **window** edit option
14812 shall be set before being used by the type character.

14813 Enter visual mode. If *type* is not specified, it shall be as if a *type* of '+' was specified. The *type*
14814 shall cause the following effects:

- + Place the beginning of the specified line at the top of the display.
- Place the end of the specified line at the bottom of the display.
- . Place the beginning of the specified line in the middle of the display.
- ^ If the specified line is less than or equal to the value of the **window** edit option, set the line
to 1; otherwise, decrement the line by the value of the **window** edit option minus 1. Place
the beginning of this line as close to the bottom of the displayed lines as possible, while still
displaying the value of the **window** edit option number of lines.

14822 *Current line*: Set to the specified line.

14823 *Current column*: Set to non-<blank>.

14824 **Write**

14825 *Synopsis:* [2addr] w[rite][!][>>][file]
14826 [2addr] w[rite][!][file]
14827 [2addr] wq[!][>>][file]

14828 If no lines are specified, the lines shall default to the entire file.

14829 The command **wq** shall be equivalent to a **write** command followed by a **quit** command; **wq!**
14830 shall be equivalent to **write!** followed by **quit**. In both cases, if the **write** command fails, the
14831 **quit** shall not be attempted.

14832 If the command name is not followed by one or more <blank>s, or *file* is not preceded by a '!'
14833 character, the **write** shall be to a file.

1. If the >> argument is specified, and the file already exists, the lines shall be appended to
the file instead of replacing its contents. If the >> argument is specified, and the file does
not already exist, it is unspecified whether the write shall proceed as if the >> argument
had not been specified or if the write shall fail.
2. If the **readonly** edit option is set (see **readonly** (on page 393)), the **write** shall fail.
3. If *file* is specified, and is not the current pathname, and the file exists, the **write** shall fail.
4. If *file* is not specified, the current pathname shall be used. If there is no current pathname,
the **write** command shall fail.
5. If the current pathname is used, and the current pathname has been changed by the **file** or
read commands, and the file exists, the **write** shall fail. If the **write** is successful,

14844 subsequent **writes** shall not fail for this reason (unless the current pathname is changed
14845 again).

14846 6. If the whole edit buffer is not being written, and the file to be written exists, the **write** shall
14847 fail.

14848 For rules 1., 2., 4., and 5., the **write** can be forced by appending the character '!' to the
14849 command name.

14850 For rules 2., 4., and 5., the **write** can be forced by setting the **writeany** edit option.

14851 Additional, implementation-defined tests may cause the **write** to fail.

14852 If the edit buffer is empty, a file without any contents shall be written.

14853 An informational message shall be written noting the number of lines and bytes written.

14854 Otherwise, if the command is followed by one or more <blank>s, and the file is preceded by
14855 '!', the rest of the line after the '!' shall have '%', '#', and '!' characters expanded as
14856 described in **Command Line Parsing in ex** (on page 360).

14857 The **ex** utility shall then pass two arguments to the program named by the **shell** edit option; the
14858 first shall be **-c** and the second shall be the expanded arguments to the **write** command as a
14859 single argument. The specified lines shall be written to the standard input of the command. The
14860 standard error and standard output of the program, if any, shall be written as described for the
14861 **print** command. If the last character in that output is not a <newline>, a <newline> shall be
14862 written at the end of the output.

14863 The special meaning of the '!' following the **write** command can be overridden by escaping it
14864 with a backslash character.

14865 *Current line:* Unchanged.

14866 *Current column:* Unchanged.

14867 Write and Exit

14868 *Synopsis:* [2addr] x[it][!][file]

14869 If the edit buffer has not been modified since the last complete **write**, **xit** shall be equivalent to
14870 the **quit** command, or if a '!' is appended to the command name, to **quit!**.

14871 Otherwise, **xit** shall be equivalent to the **wq** command, or if a '!' is appended to the command
14872 name, to **wq!**.

14873 *Current line:* Unchanged.

14874 *Current column:* Unchanged.

14875 Yank

14876 *Synopsis:* [2addr] ya[nk][buffer][count]

14877 Copy the specified lines to the specified buffer (by default, the unnamed buffer), which shall
14878 become a line-mode buffer.

14879 *Current line:* Unchanged.

14880 *Current column:* Unchanged.

14881 **Adjust Window**

14882 *Synopsis:* [*1addr*] [*z*[!][*type* . . .][*count*][*flags*]

14883 If no line is specified, the current line shall be the default; if *type* is omitted as well, the current
14884 line value shall first be incremented by 1. If incrementing the current line would cause it to be greater
14885 than the last line in the edit buffer, it shall be an error.

14886 If there are <blank>s between the *type* argument and the preceding **z** command name or optional
14887 '!' character, it shall be an error.

14888 If *count* is specified, the value of the **window** edit option shall be set to *count* (as described in
14889 **window** (on page 396)). If *count* is omitted, it shall default to 2 times the value of the **scroll** edit
14890 option, or if ! was specified, the number of lines in the display minus 1.

14891 If *type* is omitted, then *count* lines starting with the specified line shall be written. Otherwise,
14892 *count* lines starting with the line specified by the *type* argument shall be written.

14893 The *type* argument shall change the lines to be written. The possible values of *type* are as follows:

- 14894 – The specified line shall be decremented by the following value:

14895 ((number of '--' characters) x *count*) -1)

14896 If the calculation would result in a number less than 1, it shall be an error. Write lines from
14897 the edit buffer, starting at the new value of line, until *count* lines or the last line in the edit
14898 buffer has been written.

- 14899 + The specified line shall be incremented by the following value:

14900 (((number of '+-' characters) -1) x *count*) +1

14901 If the calculation would result in a number greater than the last line in the edit buffer, it
14902 shall be an error. Write lines from the edit buffer, starting at the new value of line, until
14903 *count* lines or the last line in the edit buffer has been written.

- 14904 =,. If more than a single '.' or '=' is specified, it shall be an error. The following steps shall be
14905 taken:

14906 1. If *count* is zero, nothing shall be written.

14907 2. Write as many of the *N* lines before the current line in the edit buffer as exist. If *count*
14908 or '!' was specified, *N* shall be:

14909 (*count* -1) /2

14910 Otherwise, *N* shall be:

14911 (*count* -3) /2

14912 If *N* is a number less than 3, no lines shall be written.

14913 3. If '=' was specified as the type character, write a line consisting of the smaller of the
14914 number of columns in the display divided by two, or 40 '-' characters.

14915 4. Write the current line.

14916 5. Repeat step 3.

14917 6. Write as many of the *N* lines after the current line in the edit buffer as exist. *N* shall be
14918 defined as in step 2. If *N* is a number less than 3, no lines shall be written. If *count* is
14919 less than 3, no lines shall be written.

- 14920 ^ The specified line shall be decremented by the following value:
14921 (((number of ``^'' characters) +1) x count) -1
14922 If the calculation would result in a number less than 1, it shall be an error. Write lines from
14923 the edit buffer, starting at the new value of line, until *count* lines or the last line in the edit
14924 buffer has been written.
14925 *Current line*: Set to the last line written, unless the type is =, in which case, set to the specified
14926 line.
14927 *Current column*: Set to non-<blank>.
14928 **Escape**
14929 *Synopsis:* ! *command*
14930 [*addr*]! *command*
14931 The contents of the line after the '!' shall have '%', '#', and '!' characters expanded as
14932 described in **Command Line Parsing in ex** (on page 360). If the expansion causes the text of the
14933 line to change, it shall be redisplayed, preceded by a single '!' character.
14934 The *ex* utility shall execute the program named by the **shell** edit option. It shall pass two
14935 arguments to the program; the first shall be **-c**, and the second shall be the expanded arguments
14936 to the ! command as a single argument.
14937 If no lines are specified, the standard input, standard output, and standard error of the program
14938 shall be set to the standard input, standard output, and standard error of the *ex* program when it
14939 was invoked. In addition, a warning message shall be written if the edit buffer has been
14940 modified since the last complete write, and the **warn** edit option is set.
14941 If lines are specified, they shall be passed to the program as standard input, and the standard
14942 output and standard error of the program shall replace those lines in the edit buffer. Each line in
14943 the program output (as delimited by <newline>s or the end of the output if it is not immediately
14944 preceded by a <newline>), shall be a separate line in the edit buffer. Any occurrences of
14945 <carriage-return> and <newline> pairs in the output shall be treated as single <newline>s. The
14946 specified lines shall be copied into the unnamed buffer before they are replaced, and the
14947 unnamed buffer shall become a line-mode buffer.
14948 If in *ex* mode, a single '!' character shall be written when the program completes.
14949 This command shall be affected by the **shell** and **warn** edit options. If no lines are specified, this
14950 command shall be affected by the **autowrite** and **writeany** edit options. If lines are specified, this
14951 command shall be affected by the **autoprint** edit option.
14952 *Current line*:
14953 1. If no lines are specified, unchanged.
14954 2. Otherwise, set to the last line read in, if any lines are read in.
14955 3. Otherwise, set to the line before the first line of the lines specified, if that line exists.
14956 4. Otherwise, set to the first line of the edit buffer if the edit buffer is not empty.
14957 5. Otherwise, set to zero.
14958 *Current column*: If no lines are specified, unchanged. Otherwise, set to non-<blank>.

14959 **Shift Left**

14960 *Synopsis:* [*2addr*] <[< . . .] [*count*] [*flags*]

14961 Shift the specified lines to the start of the line; the number of column positions to be shifted shall
14962 be the number of command characters times the value of the **shiftwidth** edit option. Only
14963 leading <blank>s shall be deleted or changed into other <blank>s in shifting; other characters
14964 shall not be affected.

14965 Lines to be shifted shall be copied into the unnamed buffer, which shall become a line-mode
14966 buffer.

14967 This command shall be affected by the **autoprint** edit option.

14968 *Current line:* Set to the last line in the lines specified.

14969 *Current column:* Set to non-<blank>.

14970 **Shift Right**

14971 *Synopsis:* [*2addr*] >[> . . .] [*count*] [*flags*]

14972 Shift the specified lines away from the start of the line; the number of column positions to be
14973 shifted shall be the number of command characters times the value of the **shiftwidth** edit option.
14974 The shift shall be accomplished by adding <blank>s as a prefix to the line or changing leading
14975 <blank>s into other <blank>s. Empty lines shall not be changed.

14976 Lines to be shifted shall be copied into the unnamed buffer, which shall become a line-mode
14977 buffer.

14978 This command shall be affected by the **autoprint** edit option.

14979 *Current line:* Set to the last line in the lines specified.

14980 *Current column:* Set to non-<blank>.

14981 **<control>-D**

14982 *Synopsis:* <control>-D

14983 Write the next *n* lines, where *n* is the minimum of the values of the **scroll** edit option and the
14984 number of lines after the current line in the edit buffer. If the current line is the last line of the
14985 edit buffer it shall be an error.

14986 *Current line:* Set to the last line written.

14987 *Current column:* Set to non-<blank>.

14988 **Write Line Number**

14989 *Synopsis:* [*1addr*] = [*flags*]

14990 If *line* is not specified, it shall default to the last line in the edit buffer. Write the line number of
14991 the specified line.

14992 *Current line:* Unchanged.

14993 *Current column:* Unchanged.

14994 **Execute**

14995 *Synopsis:* [*2addr*] @ *buffer*
 14996 [*2addr*] * *buffer*

14997 If no buffer is specified or is specified as '@' or '*', the last buffer executed shall be used. If no
 14998 previous buffer has been executed, it shall be an error.

14999 For each line specified by the addresses, set the current line ('.') to the specified line, and
 15000 execute the contents of the named *buffer* (as they were at the time the @ command was executed)
 15001 as ex commands. For each line of a line-mode buffer, and all but the last line of a character-mode
 15002 buffer, the ex command parser shall behave as if the line was terminated by a <newline>.

15003 If an error occurs during this process, or a line specified by the addresses does not exist when the
 15004 current line would be set to it, or more than a single line was specified by the addresses, and the
 15005 contents of the edit buffer are replaced (for example, by the ex :edit command) an error message
 15006 shall be written, and no more commands resulting from the execution of this command shall be
 15007 processed.

15008 *Current line:* As specified for the individual ex commands.

15009 *Current column:* As specified for the individual ex commands.

15010 **Regular Expressions in ex**

15011 The ex utility shall support regular expressions that are a superset of the basic regular
 15012 expressions described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic
 15013 Regular Expressions. A null regular expression ("//") shall be equivalent to the last regular
 15014 expression encountered.

15015 Regular expressions can be used in addresses to specify lines and, in some commands (for
 15016 example, the **substitute** command), to specify portions of a line to be substituted.

15017 The following constructs can be used to enhance the basic regular expressions:

15018 \< Match the beginning of a *word*. (See the definition of *word* at the beginning of **Command**
 15019 **Descriptions in ex** (on page 366).)

15020 \> Match the end of a *word*.

15021 ~ Match the replacement part of the last **substitute** command. The tilde ('~') character can
 15022 be escaped in a regular expression to become a normal character with no special meaning.
 15023 The backslash shall be discarded.

15024 When the editor option **magic** is not set, the only characters with special meanings shall be '^'
 15025 at the beginning of a pattern, '\$' at the end of a pattern, and '\'. The characters '.', '*',
 15026 '[', and '^' shall be treated as ordinary characters unless preceded by a '\'; when preceded
 15027 by a '\' they shall regain their special meaning, or in the case of backslash, be handled as a
 15028 single backslash. Backslashes used to escape other characters shall be discarded.

15029 **Replacement Strings in ex**

15030 The character '&' ('&' if the editor option **magic** is not set) in the replacement string shall
 15031 stand for the text matched by the pattern to be replaced. The character '~' ('\~' if **magic** is not
 15032 set) shall be replaced by the replacement part of the previous **substitute** command. The
 15033 sequence '\n', where *n* is an integer, shall be replaced by the text matched by the pattern
 15034 enclosed in the *n*th set of parentheses '\(' and '\)'.

15035 The strings '\l', '\u', '\L', and '\U' can be used to modify the case of elements in the
 15036 replacement string (using the '\&' or "\digit) notation. The string '\l' ('\u') shall cause

15037 the character that follows to be converted to lowercase (uppercase). The string '\L' ('\U')
15038 shall cause all characters subsequent to it to be converted to lowercase (uppercase) as they are
15039 inserted by the substitution until the string '\e' or '\E', or the end of the replacement string,
15040 is encountered.

15041 Otherwise, any character following a backslash shall be treated as that literal character, and the
15042 escaping backslash shall be discarded.

15043 An example of case conversion with the **s** command is as follows:

```
15044 :p
15045 The cat sat on the mat.
15046 :s/>\<.at\>/\u&/gp
15047 The Cat Sat on the Mat.
15048 :s/S\(.*\)M/S\U\1\em/p
15049 The Cat SAT ON THE Mat.
```

15050 Edit Options in ex

15051 The **ex** utility has a number of options that modify its behavior. These options have default
15052 settings, which can be changed using the **set** command.

15053 Options are Boolean unless otherwise specified.

15054 **autoindent, ai**

15055 [Default *unset*]

15056 If **autoindent** is set, each line in input mode shall be indented (using first as many <tab>s as
15057 possible, as determined by the editor option **tabstop**, and then using <space>s) to align with
15058 another line, as follows:

- 15059 1. If in open or visual mode and the text input is part of a line-oriented command (see the
15060 EXTENDED DESCRIPTION in **vi**), align to the first column.
- 15061 2. Otherwise, if in open or visual mode, indentation for each line shall be set as follows:
 - 15062 a. If a line was previously inserted as part of this command, it shall be set to the
15063 indentation of the last inserted line by default, or as otherwise specified for the
15064 <control>-D character in **Input Mode Commands in vi** (on page 1019).
 - 15065 b. Otherwise, it shall be set to the indentation of the previous current line, if any;
15066 otherwise, to the first column.
- 15067 3. For the **ex a, i, and c** commands, indentation for each line shall be set as follows:
 - 15068 a. If a line was previously inserted as part of this command, it shall be set to the
15069 indentation of the last inserted line by default, or as otherwise specified for the **eof**
15070 character in **Scroll** (on page 364).
 - 15071 b. Otherwise, if the command is the **ex a** command, it shall be set to the line appended
15072 after, if any; otherwise to the first column.
 - 15073 c. Otherwise, if the command is the **ex i** command, it shall be set to the line inserted
15074 before, if any; otherwise to the first column.
 - 15075 d. Otherwise, if the command is the **ex c** command, it shall be set to the indentation of
15076 the line replaced.

15077 **autoprint, ap**

15078 [Default set]

15079 If **autoprint** is set, the current line shall be written after each **ex** command that modifies the
15080 contents of the current edit buffer, and after each **tag** command for which the tag search pattern
15081 was found or tag line number was valid, unless:

- 15082 1. The command was executed while in open or visual mode.
- 15083 2. The command was executed as part of a **global** or **v** command or @ buffer execution.
- 15084 3. The command was the form of the **read** command that reads a file into the edit buffer.
- 15085 4. The command was the **append**, **change**, or **insert** command.
- 15086 5. The command was not terminated by a <newline>.
- 15087 6. The current line shall be written by a flag specified to the command; for example, **delete #**
15088 shall write the current line as specified for the flag modifier to the **delete** command, and
15089 not as specified by the **autoprint** edit option.

15090 **autowrite, aw**

15091 [Default unset]

15092 If **autowrite** is set, and the edit buffer has been modified since it was last completely written to
15093 any file, the contents of the edit buffer shall be written as if the **ex write** command had been
15094 specified without arguments, before each command affected by the **autowrite** edit option is
15095 executed. Appending the character '!' to the command name of any of the **ex** commands
15096 except '!' shall prevent the write. If the write fails, it shall be an error and the command shall
15097 not be executed.

15098 **beautify, bf**

15099 XSI [Default unset]

15100 If **beautify** is set, all non-printable characters, other than <tab>s, <newline>s, and <form-feed>s,
15101 shall be discarded from text read in from files.

15102 **directory, dir**

15103 [Default *implementation-defined*]

15104 The value of this option specifies the directory in which the editor buffer is to be placed. If this
15105 directory is not writable by the user, the editor shall quit.

15106 **edcompatible, ed**

15107 [Default unset]

15108 Causes the presence of **g** and **c** suffixes on substitute commands to be remembered, and toggled
15109 by repeating the suffixes.

15110 **errorbells, eb**

15111 [Default *unset*]

15112 If the editor is in **ex** mode, and the terminal does not support a standout mode (such as inverse video), and **errorbells** is set, error messages shall be preceded by alerting the terminal.

15114 **exrc**

15115 [Default *unset*]

15116 If **exrc** is set, **ex** shall access any **.exrc** file in the current directory, as described in **Initialization in ex and vi** (on page 356). If **exrc** is not set, **ex** shall ignore any **.exrc** file in the current directory during initialization, unless the current directory is that named by the **HOME** environment variable.

15120 **ignorecase, ic**

15121 [Default *unset*]

15122 If **ignorecase** is set, characters that have uppercase and lowercase representations shall have those representations considered as equivalent for purposes of regular expression comparison.

15124 The **ignorecase** edit option shall affect all remembered regular expressions; for example, unsetting the **ignorecase** edit option shall cause a subsequent **vi n** command to search for the last basic regular expression in a case-sensitive fashion.

15127 **list**

15128 [Default *unset*]

15129 If **list** is set, edit buffer lines written while in **ex** command mode shall be written as specified for the **print** command with the **I** flag specified. In open or visual mode, each edit buffer line shall be displayed as specified for the **ex print** command with the **I** flag specified. In open or visual text input mode, when the cursor does not rest on any character in the line, it shall rest on the '**\$**' marking the end of the line.

15134 **magic**

15135 [Default *set*]

15136 If **magic** is set, modify the interpretation of characters in regular expressions and substitution replacement strings (see **Regular Expressions in ex** (on page 389) and **Replacement Strings in ex** (on page 389)).

15139 **mesg**

15140 [Default *set*]

15141 If **mesg** is set, the permission for others to use the **write** or **talk** commands to write to the terminal shall be turned on while in open or visual mode. The shell-level command **mesg n** shall take precedence over any setting of the **ex mesg** option; that is, if **mesg y** was issued before the editor started (or in a shell escape), such as:

15145 **: !mesg y**

15146 the **mesg** option in **ex** shall suppress incoming messages, but the **mesg** option shall not enable incoming messages if **mesg n** was issued.

15148 **number, nu**

15149 [Default unset]

15150 If **number** is set, edit buffer lines written while in **ex** command mode shall be written with line
15151 numbers, in the format specified by the **print** command with the # flag specified. In **ex** text input
15152 mode, each line shall be preceded by the line number it will have in the file.

15153 In open or visual mode, each edit buffer line shall be displayed with a preceding line number, in
15154 the format specified by the **ex print** command with the # flag specified. This line number shall
15155 not be considered part of the line for the purposes of evaluating the current column; that is,
15156 column position 1 shall be the first column position after the format specified by the **print**
15157 command.

15158 **paragraphs, para**

15159 [Default in the POSIX locale `IPLPPPQPP LIpplpipbp`]

15160 The **paragraphs** edit option shall define additional paragraph boundaries for the open and visual
15161 mode commands. The **paragraphs** edit option can be set to a character string consisting of zero
15162 or more character pairs. It shall be an error to set it to an odd number of characters.

15163 **prompt**

15164 [Default set]

15165 If **prompt** is set, **ex** command mode input shall be prompted for with a colon (':'); when unset,
15166 no prompt shall be written.

15167 **readonly**

15168 [Default see text]

15169 If the **readonly** edit option is set, read-only mode shall be enabled (see **Write** (on page 384)). The
15170 **readonly** edit option shall be initialized to set if either of the following conditions are true:

- 15171 • The command-line option **-R** was specified.
- 15172 • Performing actions equivalent to the **access()** function called with the following arguments
15173 indicates that the file lacks write permission:
 - 15174 1. The current pathname is used as the *path* argument.
 - 15175 2. The constant **W_OK** is used as the *amode* argument.

15176 The **readonly** edit option may be initialized to set for other, implementation-defined reasons.
15177 The **readonly** edit option shall not be initialized to unset based on any special privileges of the
15178 user or process. The **readonly** edit option shall be reinitialized each time that the contents of the
15179 edit buffer are replaced (for example, by an **edit** or **next** command) unless the user has explicitly
15180 set it, in which case it shall remain set until the user explicitly unsets it. Once unset, it shall again
15181 be reinitialized each time that the contents of the edit buffer are replaced.

15182 **redraw**

15183 [Default *unset*]

15184 The editor simulates an intelligent terminal on a dumb terminal. (Since this is likely to require a
15185 large amount of output to the terminal, it is useful only at high transmission speeds.)

15186 **remap**

15187 [Default *set*]

15188 If **remap** is set, map translation shall allow for maps defined in terms of other maps; translation
15189 shall continue until a final product is obtained. If unset, only a one-step translation shall be done.

15190 **report**

15191 [Default 5]

15192 The value of this **report** edit option specifies what number of lines being added, copied, deleted,
15193 or modified in the edit buffer will cause an informational message to be written to the user. The
15194 following conditions shall cause an informational message. The message shall contain the
15195 number of lines added, copied, deleted, or modified, but is otherwise unspecified.

- An *ex* or *vi* editor command, other than **open**, **undo**, or **visual**, that modifies at least the value
of the **report** edit option number of lines, and which is not part of an *ex global* or **v**
command, or *ex* or *vi* buffer execution, shall cause an informational message to be written.
- An *ex yank* or *vi y* or **Y** command, that copies at least the value of the **report** edit option plus
1 number of lines, and which is not part of an *ex global* or **v** command, or *ex* or *vi* buffer
execution, shall cause an informational message to be written.
- An *ex global*, **v**, **open**, **undo**, or **visual** command or *ex* or *vi* buffer execution, that adds or
deletes a total of at least the value of the **report** edit option number of lines, and which is not
part of an *ex global* or **v** command, or *ex* or *vi* buffer execution, shall cause an informational
message to be written. (For example, if 3 lines were added and 8 lines deleted during an *ex*
visual command, 5 would be the number compared against the **report** edit option after the
command completed.)

15208 **scroll**, **scr**

15209 [Default (number of lines in the display -1)/2]

15210 The value of the **scroll** edit option shall determine the number of lines scrolled by the *ex*
15211 <control>-D and **z** commands. For the *vi* <control>-D and <control>-U commands, it shall be the
15212 initial number of lines to scroll when no previous <control>-D or <control>-U command has
15213 been executed.

15214 **sections**

15215 [Default in the POSIX locale `NHSHH HUnhsh`]

15216 The **sections** edit option shall define additional section boundaries for the open and visual mode
15217 commands. The **sections** edit option can be set to a character string consisting of zero or more
15218 character pairs; it shall be an error to set it to an odd number of characters.

15219 **shell, sh**

15220 [Default from the environment variable *SHELL*]

15221 The value of this option shall be a string. The default shall be taken from the *SHELL*
15222 environment variable. If the *SHELL* environment variable is null or empty, the *sh* (see *sh*) utility
15223 shall be the default.

15224 **shiftwidth, sw**

15225 [Default 8]

15226 The value of this option shall give the width in columns of an indentation level used during
15227 autoindentation and by the shift commands (< and >).

15228 **showmatch, sm**

15229 [Default *unset*]

15230 The functionality described for the **showmatch** edit option need not be supported on block-
15231 mode terminals or terminals with insufficient capabilities.

15232 If **showmatch** is set, in open or visual mode, when a ')' or ')' is typed, if the matching '(' or
15233 '{' is currently visible on the display, the matching '(' or '{' shall be flagged moving the
15234 cursor to its location for an unspecified amount of time.

15235 **showmode**

15236 [Default *unset*]

15237 If **showmode** is set, in open or visual mode, the current mode that the editor is in shall be
15238 displayed on the last line of the display. Command mode and text input mode shall be
15239 differentiated; other unspecified modes and implementation-defined information may be
15240 displayed.

15241 **slowopen**

15242 [Default *unset*]

15243 If **slowopen** is set during open and visual text input modes, the editor shall not update portions
15244 of the display other than those display line columns that display the characters entered by the
15245 user (see **Input Mode Commands in vi** (on page 1019)).

15246 **tabstop, ts**

15247 [Default 8]

15248 The value of this edit option shall specify the column boundary used by a <tab> in the display
15249 (see **autoprint, ap** (on page 391) and **Input Mode Commands in vi** (on page 1019)).

15250 **taglength, tl**

15251 [Default zero]

15252 The value of this edit option shall specify the maximum number of characters that are
15253 considered significant in the user-specified tag name and in the tag name from the tags file. If the
15254 value is zero, all characters in both tag names shall be significant.

15255 **tags**

15256 [Default *see text*]

15257 The value of this edit option shall be a string of <blank>-delimited pathnames of files used by
15258 the **tag** command. The default value is unspecified.

15259 **term**

15260 [Default from the environment variable *TERM*]

15261 The value of this edit option shall be a string. The default shall be taken from the *TERM* variable
15262 in the environment. If the *TERM* environment variable is empty or null, the default is
15263 unspecified. The editor shall use the value of this edit option to determine the type of the display
15264 device.

15265 The results are unspecified if the user changes the value of the term edit option after editor
15266 initialization.

15267 **terse**

15268 [Default *unset*]

15269 If **terse** is set, error messages may be less verbose. However, except for this caveat, error
15270 messages are unspecified. Furthermore, not all error messages need change for different settings
15271 of this option.

15272 **warn**

15273 [Default *set*]

15274 If **warn** is set, and the contents of the edit buffer have been modified since they were last
15275 completely written, the editor shall write a warning message before certain ! commands (see
15276 **Escape** (on page 387)).

15277 **window**

15278 [Default *see text*]

15279 A value used in open and visual mode, by the <control>-B and <control>-F commands, and, in
15280 visual mode, to specify the number of lines displayed when the screen is repainted.

15281 If the **-w** command-line option is not specified, the default value shall be set to the value of the
15282 *LINES* environment variable. If the *LINES* environment variable is empty or null, the default
15283 shall be the number of lines in the display minus 1.

15284 Setting the **window** edit option to zero or to a value greater than the number of lines in the
15285 display minus 1 (either explicitly or based on the **-w** option or the *LINES* environment variable)
15286 shall cause the **window** edit option to be set to the number of lines in the display minus 1.

15287 The baud rate of the terminal line may change the default in an implementation-defined manner.

15288 **wrapmargin, wm**
15289 [Default 0]
15290 If the value of this edit option is zero, it shall have no effect.
15291 If not in the POSIX locale, the effect of this edit option is implementation-defined.
15292 Otherwise, it shall specify a number of columns from the ending margin of the terminal.
15293 During open and visual text input modes, for each character for which any part of the character
15294 is displayed in a column that is less than **wrapmargin** columns from the ending margin of the
15295 display line, the editor shall behave as follows:
15296 1. If the character triggering this event is a <blank>, it, and all immediately preceding
15297 <blank>s on the current line entered during the execution of the current text input
15298 command, shall be discarded, and the editor shall behave as if the user had entered a single
15299 <newline> instead. In addition, if the next user-entered character is a <space>, it shall be
15300 discarded as well.
15301 2. Otherwise, if there are one or more <blank>s on the current line immediately preceding the
15302 last group of inserted non-<blank>s which was entered during the execution of the current
15303 text input command, the <blank>s shall be replaced as if the user had entered a single
15304 <newline> instead.
15305 If the **autoindent** edit option is set, and the events described in 1. or 2. are performed, any
15306 <blank>s at or after the cursor in the current line shall be discarded.
15307 The ending margin shall be determined by the system or overridden by the user, as described for
15308 **COLUMNS** in the ENVIRONMENT VARIABLES section and the Base Definitions volume of
15309 IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
15310 **wrapscan, ws**
15311 [Default *set*]
15312 If **wrapscan** is set, searches (the **ex** / or ? addresses, or open and visual mode /, ?, N, and n
15313 commands) shall wrap around the beginning or end of the edit buffer; when unset, searches
15314 shall stop at the beginning or end of the edit buffer.
15315 **writeany, wa**
15316 [Default *unset*]
15317 If **writeany** is set, some of the checks performed when executing the **ex write** commands shall be
15318 inhibited, as described in editor option **autowrite**.
15319 **EXIT STATUS**
15320 The following exit values shall be returned:
15321 0 Successful completion.
15322 >0 An error occurred.
15323 **CONSEQUENCES OF ERRORS**
15324 When any error is encountered and the standard input is not a terminal device file, **ex** shall not
15325 write the file or return to command or text input mode, and shall terminate with a non-zero exit
15326 status.
15327 Otherwise, when an unrecoverable error is encountered, it shall be equivalent to a SIGHUP
15328 asynchronous event.

15329 Otherwise, when an error is encountered, the editor shall behave as specified in **Command Line**
15330 **Parsing in ex** (on page 360).

15331 APPLICATION USAGE

15332 If a SIGSEGV signal is received while **ex** is saving a file, the file might not be successfully saved.

15333 The **next** command can accept more than one file, so usage such as:

15334 **next `ls [abc]*`**

15335 is valid; it would not be valid for the **edit** or **read** commands, for example, because they expect
15336 only one file and unspecified results occur.

15337 EXAMPLES

15338 None.

15339 RATIONALE

15340 The **ex/vi** specification is based on the historical practice found in the 4 BSD and System V
15341 implementations of **ex** and **vi**. A freely redistributable implementation of **ex/vi**, which is
15342 tracking IEEE Std 1003.1-2001 fairly closely, and demonstrates the intended changes between
15343 historical implementations and IEEE Std 1003.1-2001, may be obtained by anonymous FTP from:

15344 `ftp://ftp.rdg.opengroup/pub/mirrors/nvi`

15345 A *restricted editor* (both the historical **red** utility and modifications to **ex**) were considered and
15346 rejected for inclusion. Neither option provided the level of security that users might expect.

15347 It is recognized that **ex** visual mode and related features would be difficult, if not impossible, to
15348 implement satisfactorily on a block-mode terminal, or a terminal without any form of cursor
15349 addressing; thus, it is not a mandatory requirement that such features should work on all
15350 terminals. It is the intention, however, that an **ex** implementation should provide the full set of
15351 capabilities on all terminals capable of supporting them.

15352 Options

15353 The **-c** replacement for **+command** was inspired by the **-e** option of **sed**. Historically, all such
15354 commands (see **edit** and **next** as well) were executed from the last line of the edit buffer. This
15355 meant, for example, that "**+/pattern**" would fail unless the **wrapscan** option was set.
15356 IEEE Std 1003.1-2001 requires conformance to historical practice. Historically, some
15357 implementations restricted the **ex** commands that could be listed as part of the command line
15358 arguments. For consistency, IEEE Std 1003.1-2001 does not permit these restrictions.

15359 In historical implementations of the editor, the **-R** option (and the **readonly** edit option) only
15360 prevented overwriting of files; appending to files was still permitted, mapping loosely into the
15361 **csh noclobber** variable. Some implementations, however, have not followed this semantic, and
15362 **readonly** does not permit appending either. IEEE Std 1003.1-2001 follows the latter practice,
15363 believing that it is a more obvious and intuitive meaning of **readonly**.

15364 The **-s** option suppresses all interactive user feedback and is useful for editing scripts in batch
15365 jobs. The list of specific effects is historical practice. The terminal type "incapable of supporting
15366 open and visual modes" has historically been named "dumb".

15367 The **-t** option was required because the **ctags** utility appears in IEEE Std 1003.1-2001 and the
15368 option is available in all historical implementations of **ex**.

15369 Historically, the **ex** and **vi** utilities accepted a **-x** option, which did encryption based on the
15370 algorithm found in the historical **crypt** utility. The **-x** option for encryption, and the associated
15371 **crypt** utility, were omitted because the algorithm used was not specifiable and the export control
15372 laws of some nations make it difficult to export cryptographic technology. In addition, it did not

15373 historically provide the level of security that users might expect.

15374 Standard Input

15375 An end-of-file condition is not equivalent to an end-of-file character. A common end-of-file
15376 character, <control>-D, is historically an ex command.

15377 There was no maximum line length in historical implementations of ex. Specifically, as it was
15378 parsed in chunks, the addresses had a different maximum length than the filenames. Further, the
15379 maximum line buffer size was declared as BUFSIZ, which was different lengths on different
15380 systems. This version selected the value of {LINE_MAX} to impose a reasonable restriction on
15381 portable usage of ex and to aid test suite writers in their development of realistic tests that
15382 exercise this limit.

15383 Input Files

15384 It was an explicit decision by the standard developers that a <newline> be added to any file
15385 lacking one. It was believed that this feature of ex and vi was relied on by users in order to make
15386 text files lacking a trailing <newline> more portable. It is recognized that this will require a
15387 user-specified option or extension for implementations that permit ex and vi to edit files of type
15388 other than text if such files are not otherwise identified by the system. It was agreed that the
15389 ability to edit files of arbitrary type can be useful, but it was not considered necessary to
15390 mandate that an ex or vi implementation be required to handle files other than text files.

15391 The paragraph in the INPUT FILES section, “By default, …”, is intended to close a long-standing
15392 security problem in ex and vi; that of the “modeline” or “modelines” edit option. This feature
15393 allows any line in the first or last five lines of the file containing the strings “ex:” or “vi:”
15394 (and, apparently, “ei:” or “vx:”) to be a line containing editor commands, and ex interprets all
15395 the text up to the next ‘:’ or <newline> as a command. Consider the consequences, for
15396 example, of an unsuspecting user using ex or vi as the editor when replying to a mail message in
15397 which a line such as:

15398 ex: ! rm -rf :

15399 appeared in the signature lines. The standard developers believed strongly that an editor should
15400 not by default interpret any lines of a file. Vendors are strongly urged to delete this feature from
15401 their implementations of ex and vi.

15402 Asynchronous Events

15403 The intention of the phrase “complete write” is that the entire edit buffer be written to stable
15404 storage. The note regarding temporary files is intended for implementations that use temporary
15405 files to back edit buffers unnamed by the user.

15406 Historically, SIGQUIT was ignored by ex, but was the equivalent of the Q command in visual
15407 mode; that is, it exited visual mode and entered ex mode. IEEE Std 1003.1-2001 permits, but does
15408 not require, this behavior. Historically, SIGINT was often used by vi users to terminate text
15409 input mode (<control>-C is often easier to enter than <ESC>). Some implementations of vi
15410 alerted the terminal on this event, and some did not. IEEE Std 1003.1-2001 requires that SIGINT
15411 behave identically to <ESC>, and that the terminal not be alerted.

15412 Historically, suspending the ex editor during text input mode was similar to SIGINT, as
15413 completed lines were retained, but any partial line discarded, and the editor returned to
15414 command mode. IEEE Std 1003.1-2001 is silent on this issue; implementations are encouraged to
15415 follow historical practice, where possible.

15416 Historically, the *vi* editor did not treat SIGTSTP as an asynchronous event, and it was therefore
15417 impossible to suspend the editor in visual text input mode. There are two major reasons for this.
15418 The first is that SIGTSTP is a broadcast signal on UNIX systems, and the chain of events where
15419 the shell execs an application that then execs *vi* usually caused confusion for the terminal state if
15420 SIGTSTP was delivered to the process group in the default manner. The second was that most
15421 implementations of the UNIX *curses* package are not reentrant, and the receipt of SIGTSTP at the
15422 wrong time will cause them to crash. IEEE Std 1003.1-2001 is silent on this issue; implementations
15423 are encouraged to treat suspension as an asynchronous event if possible.

15424 Historically, modifications to the edit buffer made before SIGINT interrupted an operation were
15425 retained; that is, anywhere from zero to all of the lines to be modified might have been modified
15426 by the time the SIGINT arrived. These changes were not discarded by the arrival of SIGINT.
15427 IEEE Std 1003.1-2001 permits this behavior, noting that the **undo** command is required to be able
15428 to undo these partially completed commands.

15429 The action taken for signals other than SIGINT, SIGCONT, SIGHUP, and SIGTERM is
15430 unspecified because some implementations attempt to save the edit buffer in a useful state when
15431 other signals are received.

15432 Standard Error

15433 For *ex/vi*, diagnostic messages are those messages reported as a result of a failed attempt to
15434 invoke *ex* or *vi*, such as invalid options or insufficient resources, or an abnormal termination
15435 condition. Diagnostic messages should not be confused with the error messages generated by
15436 inappropriate or illegal user commands.

15437 Initialization in ex and vi

15438 If an *ex* command (other than **cd**, **chdir**, or **source**) has a filename argument, one or both of the
15439 alternate and current pathnames will be set. Informally, they are set as follows:

- 15440 1. If the *ex* command is one that replaces the contents of the edit buffer, and it succeeds, the
15441 current pathname will be set to the filename argument (the first filename argument in the
15442 case of the **next** command) and the alternate pathname will be set to the previous current
15443 pathname, if there was one.
- 15444 2. In the case of the file read/write forms of the **read** and **write** commands, if there is no
15445 current pathname, the current pathname will be set to the filename argument.
- 15446 3. Otherwise, the alternate pathname will be set to the filename argument.

15447 For example, **:edit foo** and **:recover foo**, when successful, set the current pathname, and, if there
15448 was a previous current pathname, the alternate pathname. The commands **:write**, **!command**,
15449 and **:edit** set neither the current or alternate pathnames. If the **:edit foo** command were to fail for
15450 some reason, the alternate pathname would be set. The **read** and **write** commands set the
15451 alternate pathname to their *file* argument, unless the current pathname is not set, in which case
15452 they set the current pathname to their *file* arguments. The alternate pathname was not
15453 historically set by the **:source** command. IEEE Std 1003.1-2001 requires conformance to historical
15454 practice. Implementations adding commands that take filenames as arguments are encouraged
15455 to set the alternate pathname as described here.

15456 Historically, *ex* and *vi* read the **.exrc** file in the **\$HOME** directory twice, if the editor was executed
15457 in the **\$HOME** directory. IEEE Std 1003.1-2001 prohibits this behavior.

15458 Historically, the 4 BSD *ex* and *vi* read the **\$HOME** and local **.exrc** files if they were owned by the
15459 real ID of the user, or the **sourceany** option was set, regardless of other considerations. This was
15460 a security problem because it is possible to put normal UNIX system commands inside a **.exrc**

15461 file. IEEE Std 1003.1-2001 does not specify the **sourceany** option, and historical implementations
15462 are encouraged to delete it.

15463 The **.exrc** files must be owned by the real ID of the user, and not writable by anyone other than
15464 the owner. The appropriate privileges exception is intended to permit users to acquire special
15465 privileges, but continue to use the **.exrc** files in their home directories.

15466 System V Release 3.2 and later *vi* implementations added the option [**no**]**exrc**. The behavior is
15467 that local **.exrc** files are read-only if the **exrc** option is set. The default for the **exrc** option was off,
15468 so by default, local **.exrc** files were not read. The problem this was intended to solve was that
15469 System V permitted users to give away files, so there is no possible ownership or writeability
15470 test to ensure that the file is safe. This is still a security problem on systems where users can give
15471 away files, but there is nothing additional that IEEE Std 1003.1-2001 can do. The
15472 implementation-defined exception is intended to permit groups to have local **.exrc** files that are
15473 shared by users, by creating pseudo-users to own the shared files.

15474 IEEE Std 1003.1-2001 does not mention system-wide *ex* and *vi* start-up files. While they exist in
15475 several implementations of *ex* and *vi*, they are not present in any implementations considered
15476 historical practice by IEEE Std 1003.1-2001. Implementations that have such files should use
15477 them only if they are owned by the real user ID or an appropriate user (for example, root on
15478 UNIX systems) and if they are not writable by any user other than their owner. System-wide
15479 start-up files should be read before the *EXINIT* variable, **\$HOME/.exrc**, or local **.exrc** files are
15480 evaluated.

15481 Historically, any *ex* command could be entered in the *EXINIT* variable or the **.exrc** file, although
15482 ones requiring that the edit buffer already contain lines of text generally caused historical
15483 implementations of the editor to drop **core**. IEEE Std 1003.1-2001 requires that any *ex* command
15484 be permitted in the *EXINIT* variable and **.exrc** files, for simplicity of specification and
15485 consistency, although many of them will obviously fail under many circumstances.

15486 The initialization of the contents of the edit buffer uses the phrase “the effect shall be” with
15487 regard to various *ex* commands. The intent of this phrase is that edit buffer contents loaded
15488 during the initialization phase not be lost; that is, loading the edit buffer should fail if the **.exrc**
15489 file read in the contents of a file and did not subsequently write the edit buffer. An additional
15490 intent of this phrase is to specify that the initial current line and column is set as specified for the
15491 individual *ex* commands.

15492 Historically, the **-t** option behaved as if the tag search were a **+command**; that is, it was executed
15493 from the last line of the file specified by the tag. This resulted in the search failing if the pattern
15494 was a forward search pattern and the **wrapscan** edit option was not set. IEEE Std 1003.1-2001
15495 does not permit this behavior, requiring that the search for the tag pattern be performed on the
15496 entire file, and, if not found, that the current line be set to a more reasonable location in the file.

15497 Historically, the empty edit buffer presented for editing when a file was not specified by the user
15498 was unnamed. This is permitted by IEEE Std 1003.1-2001; however, implementations are
15499 encouraged to provide users a temporary filename for this buffer because it permits them the
15500 use of *ex* commands that use the current pathname during temporary edit sessions.

15501 Historically, the file specified using the **-t** option was not part of the current argument list. This
15502 practice is permitted by IEEE Std 1003.1-2001; however, implementations are encouraged to
15503 include its name in the current argument list for consistency.

15504 Historically, the **-c** command was generally not executed until a file that already exists was
15505 edited. IEEE Std 1003.1-2001 requires conformance to this historical practice. Commands that
15506 could cause the **-c** command to be executed include the *ex* commands **edit**, **next**, **recover**,
15507 **rewind**, and **tag**, and the *vi* commands **<control>-^** and **<control>-J**. Historically, reading a file
15508 into an edit buffer did not cause the **-c** command to be executed (even though it might set the

15509 current pathname) with the exception that it did cause the **-c** command to be executed if: the
15510 editor was in **ex** mode, the edit buffer had no current pathname, the edit buffer was empty, and
15511 no read commands had yet been attempted. For consistency and simplicity of specification,
15512 IEEE Std 1003.1-2001 does not permit this behavior.

15513 Historically, the **-r** option was the same as a normal edit session if there was no recovery
15514 information available for the file. This allowed users to enter:

15515 `vi -r *.c`

15516 and recover whatever files were recoverable. In some implementations, recovery was attempted
15517 only on the first file named, and the file was not entered into the argument list; in others,
15518 recovery was attempted for each file named. In addition, some historical implementations
15519 ignored **-r** if **-t** was specified or did not support command line *file* arguments with the **-t** option.
15520 For consistency and simplicity of specification, IEEE Std 1003.1-2001 disallows these special
15521 cases, and requires that recovery be attempted the first time each file is edited.

15522 Historically, **vi** initialized the ‘ and ’ marks, but **ex** did not. This meant that if the first command
15523 in **ex** mode was **visual** or if an **ex** command was executed first (for example, **vi +10 file**), **vi** was
15524 entered without the marks being initialized. Because the standard developers believed the marks
15525 to be generally useful, and for consistency and simplicity of specification, IEEE Std 1003.1-2001
15526 requires that they always be initialized if in open or visual mode, or if in **ex** mode and the edit
15527 buffer is not empty. Not initializing it in **ex** mode if the edit buffer is empty is historical practice;
15528 however, it has always been possible to set (and use) marks in empty edit buffers in open and
15529 visual mode edit sessions.

15530 Addressing

15531 Historically, **ex** and **vi** accepted the additional addressing forms ‘\//’ and ‘\?’. They were
15532 equivalent to “//” and “??”, respectively. They are not required by IEEE Std 1003.1-2001,
15533 mostly because nobody can remember whether they ever did anything different historically.

15534 Historically, **ex** and **vi** permitted an address of zero for several commands, and permitted the %
15535 address in empty files for others. For consistency, IEEE Std 1003.1-2001 requires support for the
15536 former in the few commands where it makes sense, and disallows it otherwise. In addition,
15537 because IEEE Std 1003.1-2001 requires that % be logically equivalent to “1,\$”, it is also
15538 supported where it makes sense and disallowed otherwise.

15539 Historically, the % address could not be followed by further addresses. For consistency and
15540 simplicity of specification, IEEE Std 1003.1-2001 requires that additional addresses be supported.

15541 All of the following are valid *addresses*:

15542 `+++` Three lines after the current line.

15543 `/re/-` One line before the next occurrence of *re*.

15544 `-2` Two lines before the current line.

15545 `3 ----- 2` Line one (note intermediate negative address).

15546 `1 2 3` Line six.

15547 Any number of addresses can be provided to commands taking addresses; for example,
15548 “1,2,3,4,5P” prints lines 4 and 5, because two is the greatest valid number of addresses
15549 accepted by the **print** command. This, in combination with the semicolon delimiter, permits
15550 users to create commands based on ordered patterns in the file. For example, the command
15551 **3;/foo;/+2print** will display the first line after line 3 that contains the pattern *foo*, plus the next
15552 two lines. Note that the address **3;** must be evaluated before being discarded because the search

15553 origin for the **/foo/** command depends on this.

15554 Historically, values could be added to addresses by including them after one or more <blank>s;

15555 for example, **3 – 5p** wrote the seventh line of the file, and **/foo/ 5** was the same as **/foo/+5**.

15556 However, only absolute values could be added; for example, **5 /foo/** was an error.

15557 IEEE Std 1003.1-2001 requires conformance to historical practice. Address offsets are separately

15558 specified from addresses because they could historically be provided to visual mode search

15559 commands.

15560 Historically, any missing addresses defaulted to the current line. This was true for leading and

15561 trailing comma-delimited addresses, and for trailing semicolon-delimited addresses. For

15562 consistency, IEEE Std 1003.1-2001 requires it for leading semicolon addresses as well.

15563 Historically, **ex** and **vi** accepted the **'^'** character as both an address and as a flag offset for

15564 commands. In both cases it was identical to the **'-'** character. IEEE Std 1003.1-2001 does not

15565 require or prohibit this behavior.

15566 Historically, the enhancements to basic regular expressions could be used in addressing; for

15567 example, **'~'**, **'\<'**, and **'\>'**. IEEE Std 1003.1-2001 requires conformance to historical

15568 practice; that is, that regular expression usage be consistent, and that regular expression

15569 enhancements be supported wherever regular expressions are used.

15570 Command Line Parsing in ex

15571 Historical **ex** command parsing was even more complex than that described here.

15572 IEEE Std 1003.1-2001 requires the subset of the command parsing that the standard developers

15573 believed was documented and that users could reasonably be expected to use in a portable

15574 fashion, and that was historically consistent between implementations. (The discarded

15575 functionality is obscure, at best.) Historical implementations will require changes in order to

15576 comply with IEEE Std 1003.1-2001; however, users are not expected to notice any of these

15577 changes. Most of the complexity in **ex** parsing is to handle three special termination cases:

- 15578 1. The **!, global, v**, and the filter versions of the **read** and **write** commands are delimited by
15579 <newline>s (they can contain vertical-line characters that are usually shell pipes).
- 15580 2. The **ex, edit, next, and visual** in open and visual mode commands all take **ex** commands,
15581 optionally containing vertical-line characters, as their first arguments.
- 15582 3. The **s** command takes a regular expression as its first argument, and uses the delimiting
15583 characters to delimit the command.

15584 Historically, vertical-line characters in the **+command** argument of the **ex, edit, next, vi,** and

15585 **visual** commands, and in the **pattern** and **replacement** parts of the **s** command, did not delimit the

15586 command, and in the filter cases for **read** and **write**, and the **!, global**, and **v** commands, they did

15587 not delimit the command at all. For example, the following commands are all valid:

```
15588 :edit +25 | s/abc/ABC/ file.c
15589 :s/ | /PIPE/
15590 :read !spell % | colummate
15591 :global/pattern/p | l
15592 :s/a/b/ | s/c/d | set
```

15593 Historically, empty or <blank> filled lines in **.exrc** files and **sourced** files (as well as **EXINIT**

15594 variables and **ex** command scripts) were treated as default commands; that is, **print** commands.

15595 IEEE Std 1003.1-2001 specifically requires that they be ignored when encountered in **.exrc** and

15596 **sourced** files to eliminate a common source of new user error.

15597 Historically, *ex* commands with multiple adjacent (or <blank>-separated) vertical lines were
15598 handled oddly when executed from *ex* mode. For example, the command ||| <carriage-return>,
15599 when the cursor was on line 1, displayed lines 2, 3, and 5 of the file. In addition, the command |
15600 would only display the line after the next line, instead of the next two lines. The former worked
15601 more logically when executed from *vi* mode, and displayed lines 2, 3, and 4.
15602 IEEE Std 1003.1-2001 requires the *vi* behavior; that is, a single default command and line number
15603 increment for each command separator, and trailing <newline>s after vertical-line separators are
15604 discarded.

15605 Historically, *ex* permitted a single extra colon as a leading command character; for example,
15606 :g/pattern:/p was a valid command. IEEE Std 1003.1-2001 generalizes this to require that any
15607 number of leading colon characters be stripped.

15608 Historically, any prefix of the **delete** command could be followed without intervening <blank>s
15609 by a flag character because in the command d p, p is interpreted as the buffer p.
15610 IEEE Std 1003.1-2001 requires conformance to historical practice.

15611 Historically, the **k** command could be followed by the mark name without intervening
15612 <blank>s. IEEE Std 1003.1-2001 requires conformance to historical practice.

15613 Historically, the **s** command could be immediately followed by flag and option characters; for
15614 example, s/e/E/|s|sgc3p was a valid command. However, flag characters could not stand alone;
15615 for example, the commands sp and s l would fail, while the command sgp and s gl would
15616 succeed. (Obviously, the '#' flag character was used as a delimiter character if it followed the
15617 command.) Another issue was that option characters had to precede flag characters even when
15618 the command was fully specified; for example, the command s/e/E/pg would fail, while the
15619 command s/e/E/gp would succeed. IEEE Std 1003.1-2001 requires conformance to historical
15620 practice.

15621 Historically, the first command name that had a prefix matching the input from the user was the
15622 executed command; for example, ve, ver, and vers all executed the **version** command.
15623 Commands were in a specific order, however, so that a matched **append**, not **abbreviate**.
15624 IEEE Std 1003.1-2001 requires conformance to historical practice. The restriction on command
15625 search order for implementations with extensions is to avoid the addition of commands such
15626 that the historical prefixes would fail to work portably.

15627 Historical implementations of *ex* and *vi* did not correctly handle multiple *ex* commands,
15628 separated by vertical-line characters, that entered or exited visual mode or the editor. Because
15629 implementations of *vi* exist that do not exhibit this failure mode, IEEE Std 1003.1-2001 does not
15630 permit it.

15631 The requirement that alphabetic command names consist of all following alphabetic characters
15632 up to the next non-alphabetic character means that alphabetic command names must be
15633 separated from their arguments by one or more non-alphabetic characters, normally a <blank>
15634 or '!' character, except as specified for the exceptions, the **delete**, **k**, and **s** commands.

15635 Historically, the repeated execution of the *ex* default **print** commands (<control>-D, eof,
15636 <newline>, <carriage-return>) erased any prompting character and displayed the next lines
15637 without scrolling the terminal; that is, immediately below any previously displayed lines. This
15638 provided a cleaner presentation of the lines in the file for the user. IEEE Std 1003.1-2001 does not
15639 require this behavior because it may be impossible in some situations; however,
15640 implementations are strongly encouraged to provide this semantic if possible.

15641 Historically, it was possible to change files in the middle of a command, and have the rest of the
15642 command executed in the new file; for example:

```
15643 :edit +25 file.c | s/abc/ABC/ | 1
15644 was a valid command, and the substitution was attempted in the newly edited file.
15645 IEEE Std 1003.1-2001 requires conformance to historical practice. The following commands are
15646 examples that exercise the ex parser:
15647 echo 'foo | bar' > file1; echo 'foo/bar' > file2;
15648 vi
15649 :edit +1 | s/|/PIPE/ | w file1 | e file2 | 1 | s/\//SLASH/ | wq
15650 Historically, there was no protection in editor implementations to avoid ex global, v, @, or *
15651 commands changing edit buffers during execution of their associated commands. Because this
15652 would almost invariably result in catastrophic failure of the editor, and implementations exist
15653 that do exhibit these problems, IEEE Std 1003.1-2001 requires that changing the edit buffer
15654 during a global or v command, or during a @ or * command for which there will be more than a
15655 single execution, be an error. Implementations supporting multiple edit buffers simultaneously
15656 are strongly encouraged to apply the same semantics to switching between buffers as well.
15657 The ex command quoting required by IEEE Std 1003.1-2001 is a superset of the quoting in
15658 historical implementations of the editor. For example, it was not historically possible to escape a
15659 <blank> in a filename; for example, :edit foo\\ bar would report that too many filenames had
15660 been entered for the edit command, and there was no method of escaping a <blank> in the first
15661 argument of an edit, ex, next, or visual command at all. IEEE Std 1003.1-2001 extends historical
15662 practice, requiring that quoting behavior be made consistent across all ex commands, except for
15663 the map, unmap, abbreviate, and unabbreviate commands, which historically used <control>-V
15664 instead of backslashes for quoting. For those four commands, IEEE Std 1003.1-2001 requires
15665 conformance to historical practice.
15666 Backslash quoting in ex is non-intuitive. Backslash escapes are ignored unless they escape a
15667 special character; for example, when performing file argument expansion, the string "\%\%" is
15668 equivalent to '\%', not "\<current pathname>". This can be confusing for users because
15669 backslash is usually one of the characters that causes shell expansion to be performed, and
15670 therefore shell quoting rules must be taken into consideration. Generally, quoting characters are
15671 only considered if they escape a special character, and a quoting character must be provided for
15672 each layer of parsing for which the character is special. As another example, only a single
15673 backslash is necessary for the '\1' sequence in substitute replacement patterns, because the
15674 character '1' is not special to any parsing layer above it.
15675 <control>-V quoting in ex is slightly different from backslash quoting. In the four commands
15676 where <control>-V quoting applies (abbreviate, unabbreviate, map, and unmap), any character
15677 may be escaped by a <control>-V whether it would have a special meaning or not.
15678 IEEE Std 1003.1-2001 requires conformance to historical practice.
15679 Historical implementations of the editor did not require delimiters within character classes to be
15680 escaped; for example, the command :s/[]/ on the string "xxx/yyy" would delete the '/' from
15681 the string. IEEE Std 1003.1-2001 disallows this historical practice for consistency and because it
15682 places a large burden on implementations by requiring that knowledge of regular expressions be
15683 built into the editor parser.
15684 Historically, quoting <newline>s in ex commands was handled inconsistently. In most cases, the
15685 <newline> always terminated the command, regardless of any preceding escape character,
15686 because backslash characters did not escape <newline>s for most ex commands. However, some
15687 ex commands (for example, s, map, and abbreviation) permitted <newline>s to be escaped
15688 (although in the case of map and abbreviation, <control>-V characters escaped them instead of
15689 backslashes). This was true in not only the command line, but also .exrc and sourced files. For
15690 example, the command:
```

15691 map = foo<control-V><newline>bar
15692 would succeed, although it was sometimes difficult to get the <control>-V and the inserted
15693 <newline> passed to the *ex* parser. For consistency and simplicity of specification,
15694 IEEE Std 1003.1-2001 requires that it be possible to escape <newline>s in *ex* commands at all
15695 times, using backslashes for most *ex* commands, and using <control>-V characters for the **map**
15696 and **abbreviation** commands. For example, the command **print<newline>list** is required to be
15697 parsed as the single command **print<newline>list**. While this differs from historical practice,
15698 IEEE Std 1003.1-2001 developers believed it unlikely that any script or user depended on the
15699 historical behavior.

15700 Historically, an error in a command specified using the **-c** option did not cause the rest of the **-c**
15701 commands to be discarded. IEEE Std 1003.1-2001 disallows this for consistency with mapped
15702 keys, the **@**, **global**, **source**, and **v** commands, the **EXINIT** environment variable, and the **.exrc**
15703 files.

15704 **Input Editing in ex**

15705 One of the common uses of the historical *ex* editor is over slow network connections. Editors
15706 that run in canonical mode can require far less traffic to and from, and far less processing on, the
15707 host machine, as well as more easily supporting block-mode terminals. For these reasons,
15708 IEEE Std 1003.1-2001 requires that *ex* be implemented using canonical mode input processing, as
15709 was done historically.

15710 IEEE Std 1003.1-2001 does not require the historical 4 BSD input editing characters “word erase”
15711 or “literal next”. For this reason, it is unspecified how they are handled by *ex*, although they
15712 must have the required effect. Implementations that resolve them after the line has been ended
15713 using a <newline> or <control>-M character, and implementations that rely on the underlying
15714 system terminal support for this processing, are both conforming. Implementations are strongly
15715 urged to use the underlying system functionality, if at all possible, for compatibility with other
15716 system text input interfaces.

15717 Historically, when the **eof** character was used to decrement the **autoindent** level, the cursor
15718 moved to display the new end of the **autoindent** characters, but did not move the cursor to a
15719 new line, nor did it erase the <control>-D character from the line. IEEE Std 1003.1-2001 does not
15720 specify that the cursor remain on the same line or that the rest of the line is erased; however,
15721 implementations are strongly encouraged to provide the best possible user interface; that is, the
15722 cursor should remain on the same line, and any <control>-D character on the line should be
15723 erased.

15724 IEEE Std 1003.1-2001 does not require the historical 4 BSD input editing character “reprint”,
15725 traditionally <control>-R, which redisplayed the current input from the user. For this reason,
15726 and because the functionality cannot be implemented after the line has been terminated by the
15727 user, IEEE Std 1003.1-2001 makes no requirements about this functionality. Implementations are
15728 strongly urged to make this historical functionality available, if possible.

15729 Historically, <control>-Q did not perform a literal next function in *ex*, as it did in *vi*.
15730 IEEE Std 1003.1-2001 requires conformance to historical practice to avoid breaking historical *ex*
15731 scripts and **.exrc** files.

15732

eof

15733

Whether the *eof* character immediately modifies the **autoindent** characters in the prompt is left unspecified so that implementations can conform in the presence of systems that do not support this functionality. Implementations are encouraged to modify the line and redisplay it immediately, if possible.

15737

The specification of the handling of the *eof* character differs from historical practice only in that *eof* characters are not discarded if they follow normal characters in the text input. Historically, they were always discarded.

15740

Command Descriptions in ex

15741

Historically, several commands (for example, **global**, **v**, **visual**, **s**, **write**, **wq**, **yank**, **!**, **<**, **>**, **&**, and **¬**) were executable in empty files (that is, the default address(es) were 0), or permitted explicit addresses of 0 (for example, 0 was a valid address, or 0,0 was a valid range). Addresses of 0, or command execution in an empty file, make sense only for commands that add new text to the edit buffer or write commands (because users may wish to write empty files). IEEE Std 1003.1-2001 requires this behavior for such commands and disallows it otherwise, for consistency and simplicity of specification.

15748

A count to an *ex* command has been historically corrected to be no greater than the last line in a file; for example, in a five-line file, the command **1,6print** would fail, but the command **1print300** would succeed. IEEE Std 1003.1-2001 requires conformance to historical practice.

15751

Historically, the use of flags in *ex* commands could be obscure. General historical practice was as described by IEEE Std 1003.1-2001, but there were some special cases. For instance, the **list**, **number**, and **print** commands ignored trailing address offsets; for example, **3p +++#** would display line 3, and 3 would be the current line after the execution of the command. The **open** and **visual** commands ignored both the trailing offsets and the trailing flags. Also, flags specified to the **open** and **visual** commands interacted badly with the **list** edit option, and setting and then unsetting it during the open/visual session would cause *vi* to stop displaying lines in the specified format. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit any of these exceptions to the general rule.

15760

IEEE Std 1003.1-2001 uses the word *copy* in several places when discussing buffers. This is not intended to imply implementation.

15762

Historically, *ex* users could not specify numeric buffers because of the ambiguity this would cause; for example, in the command **3 delete 2**, it is unclear whether 2 is a buffer name or a *count*. IEEE Std 1003.1-2001 requires conformance to historical practice by default, but does not preclude extensions.

15766

Historically, the contents of the unnamed buffer were frequently discarded after commands that did not explicitly affect it; for example, when using the **edit** command to switch files. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.

15769

The *ex* utility did not historically have access to the numeric buffers, and, furthermore, deleting lines in *ex* did not modify their contents. For example, if, after doing a delete in *vi*, the user switched to *ex*, did another delete, and then switched back to *vi*, the contents of the numeric buffers would not have changed. IEEE Std 1003.1-2001 requires conformance to historical practice. Numeric buffers are described in the *ex* utility in order to confine the description of buffers to a single location in IEEE Std 1003.1-2001.

15775

The metacharacters that trigger shell expansion in *file* arguments match historical practice, as does the method for doing shell expansion. Implementations wishing to provide users with the flexibility to alter the set of metacharacters are encouraged to provide a **shellmeta** string edit

15778 option.

15779 Historically, ex commands executed from vi refreshed the screen when it did not strictly need to
15780 do so; for example, **!date > /dev/null** does not require a screen refresh because the output of the
15781 UNIX *date* command requires only a single line of the screen. IEEE Std 1003.1-2001 requires that
15782 the screen be refreshed if it has been overwritten, but makes no requirements as to how an
15783 implementation should make that determination. Implementations may prompt and refresh the
15784 screen regardless.

15785 Abbreviate

15786 Historical practice was that characters that were entered as part of an abbreviation replacement
15787 were subject to **map** expansions, the **showmatch** edit option, further abbreviation expansions,
15788 and so on; that is, they were logically pushed onto the terminal input queue, and were not a
15789 simple replacement. IEEE Std 1003.1-2001 requires conformance to historical practice. Historical
15790 practice was that whenever a non-word character (that had not been escaped by a <control>-V)
15791 was entered after a word character, vi would check for abbreviations. The check was based on
15792 the type of the character entered before the word character of the word/non-word pair that
15793 triggered the check. The word character of the word/non-word pair that triggered the check and
15794 all characters entered before the trigger pair that were of that type were included in the check,
15795 with the exception of <blank>s, which always delimited the abbreviation.

15796 This means that, for the abbreviation to work, the *lhs* must end with a word character, there can
15797 be no transitions from word to non-word characters (or *vice versa*) other than between the last
15798 and next-to-last characters in the *lhs*, and there can be no <blank>s in the *lhs*. In addition,
15799 because of the historical quoting rules, it was impossible to enter a literal <control>-V in the *lhs*.
15800 IEEE Std 1003.1-2001 requires conformance to historical practice. Historical implementations did
15801 not inform users when abbreviations that could never be used were entered; implementations
15802 are strongly encouraged to do so.

15803 For example, the following abbreviations will work:

15804 :ab (p REPLACE
15805 :ab p REPLACE
15806 :ab ((p REPLACE

15807 The following abbreviations will not work:

15808 :ab (REPLACE
15809 :ab (pp REPLACE

15810 Historical practice is that words on the vi colon command line were subject to abbreviation
15811 expansion, including the arguments to the **abbrev** (and more interestingly) the **unabbrev**
15812 command. Because there are implementations that do not do abbreviation expansion for the first
15813 argument to those commands, this is permitted, but not required, by IEEE Std 1003.1-2001.
15814 However, the following sequence:

15815 :ab foo bar
15816 :ab foo baz

15817 resulted in the addition of an abbreviation of "baz" for the string "bar" in historical ex/vi, and
15818 the sequence:

15819 :ab fool bar
15820 :ab foo2 bar
15821 :unabbreviate foo2

15822 deleted the abbreviation "foo1", not "foo2". These behaviors are not permitted by
15823 IEEE Std 1003.1-2001 because they clearly violate the expectations of the user.

15824 It was historical practice that <control>-V, not backslash, characters be interpreted as escaping
15825 subsequent characters in the **abbreviate** command. IEEE Std 1003.1-2001 requires conformance
15826 to historical practice; however, it should be noted that an abbreviation containing a <blank> will
15827 never work.

15828 Append

15829 Historically, any text following a vertical-line command separator after an **append**, **change**, or
15830 **insert** command became part of the insert text. For example, in the command:

15831 :g/pattern/append|stuff1

15832 a line containing the text "stuff1" would be appended to each line matching pattern. It was
15833 also historically valid to enter:

15834 :append|stuff1

15835 stuff2

15836 .

15837 and the text on the *ex* command line would be appended along with the text inserted after it.
15838 There was an historical bug, however, that the user had to enter two terminating lines (the ' . ' lines)
15839 to terminate text input mode in this case. IEEE Std 1003.1-2001 requires conformance to
15840 historical practice, but disallows the historical need for multiple terminating lines.

15841 Change

15842 See the RATIONALE for the **append** command. Historical practice for cursor positioning after
15843 the change command when no text is input, is as described in IEEE Std 1003.1-2001. However,
15844 one System V implementation is known to have been modified such that the cursor is positioned
15845 on the first address specified, and not on the line before the first address. IEEE Std 1003.1-2001
15846 disallows this modification for consistency.

15847 Historically, the **change** command did not support buffer arguments, although some
15848 implementations allow the specification of an optional buffer. This behavior is neither required
15849 nor disallowed by IEEE Std 1003.1-2001.

15850 Change Directory

15851 A common extension in *ex* implementations is to use the elements of a **cdpath** edit option as
15852 prefix directories for *path* arguments to **chdir** that are relative pathnames and that do not have
15853 ' .' or " .. " as their first component. Elements in the **cdpath** edit option are colon-separated.
15854 The initial value of the **cdpath** edit option is the value of the shell *CDPATH* environment
15855 variable. This feature was not included in IEEE Std 1003.1-2001 because it does not exist in any
15856 of the implementations considered historical practice.

15857 Copy

15858 Historical implementations of *ex* permitted copies to lines inside of the specified range; for
15859 example, :2,5copy3 was a valid command. IEEE Std 1003.1-2001 requires conformance to
15860 historical practice.

15861

Delete

15862

IEEE Std 1003.1-2001 requires support for the historical parsing of a **delete** command followed by flags, without any intervening <blank>s. For example:

15863

1dp Deletes the first line and prints the line that was second.

15864

1deleP As for **1dp**.

15865

1d Deletes the first line, saving it in buffer *p*.

15866

1d p1l (Pee-one-ell.) Deletes the first line, saving it in buffer *p*, and listing the line that was second.

15867

Edit

15868

Historically, any *ex* command could be entered as a *+command* argument to the **edit** command, although some (for example, **insert** and **append**) were known to confuse historical implementations. For consistency and simplicity of specification, IEEE Std 1003.1-2001 requires that any command be supported as an argument to the **edit** command.

15869

Historically, the *command* argument was executed with the current line set to the last line of the file, regardless of whether the **edit** command was executed from visual mode or not. IEEE Std 1003.1-2001 requires conformance to historical practice.

15870

Historically, the *+command* specified to the **edit** and **next** commands was delimited by the first <blank>, and there was no way to quote them. For consistency, IEEE Std 1003.1-2001 requires that the usual *ex* backslash quoting be provided.

15871

Historically, specifying the *+command* argument to the **edit** command required a filename to be specified as well; for example, **:edit +100** would always fail. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this usage to fail for that reason.

15872

Historically, only the cursor position of the last file edited was remembered by the editor. IEEE Std 1003.1-2001 requires that this be supported; however, implementations are permitted to remember and restore the cursor position for any file previously edited.

15873

File

15874

Historical versions of the *ex* editor **file** command displayed a current line and number of lines in the edit buffer of 0 when the file was empty, while the *vi* <control>-G command displayed a current line and number of lines in the edit buffer of 1 in the same situation. IEEE Std 1003.1-2001 does not permit this discrepancy, instead requiring that a message be displayed indicating that the file is empty.

15875

Global

15876

The two-pass operation of the **global** and **v** commands is not intended to imply implementation, only the required result of the operation.

15877

The current line and column are set as specified for the individual *ex* commands. This requirement is cumulative; that is, the current line and column must track across all the commands executed by the **global** or **v** commands.

15898 **Insert**

15899 See the RATIONALE for the **append** command.

15900 Historically, **insert** could not be used with an address of zero; that is, not when the edit buffer
15901 was empty. IEEE Std 1003.1-2001 requires that this command behave consistently with the
15902 **append** command.

15903 **Join**

15904 The action of the **join** command in relation to the special characters is only defined for the
15905 POSIX locale because the correct amount of white space after a period varies; in Japanese none is
15906 required, in French only a single space, and so on.

15907 **List**

15908 The historical output of the **list** command was potentially ambiguous. The standard developers
15909 believed correcting this to be more important than adhering to historical practice, and
15910 IEEE Std 1003.1-2001 requires unambiguous output.

15911 **Map**

15912 Historically, command mode maps only applied to command names; for example, if the
15913 character 'x' was mapped to 'y', the command fx searched for the 'x' character, not the 'y'
15914 character. IEEE Std 1003.1-2001 requires this behavior. Historically, entering <control>-V as the
15915 first character of a vi command was an error. Several implementations have extended the
15916 semantics of vi such that <control>-V means that the subsequent command character is not
15917 mapped. This is permitted, but not required, by IEEE Std 1003.1-2001. Regardless, using
15918 <control>-V to escape the second or later character in a sequence of characters that might match
15919 a **map** command, or any character in text input mode, is historical practice, and stops the entered
15920 keys from matching a map. IEEE Std 1003.1-2001 requires conformance to historical practice.

15921 Historical implementations permitted digits to be used as a **map** command *lhs*, but then ignored
15922 the map. IEEE Std 1003.1-2001 requires that the mapped digits not be ignored.

15923 The historical implementation of the **map** command did not permit **map** commands that were
15924 more than a single character in length if the first character was printable. This behavior is
15925 permitted, but not required, by IEEE Std 1003.1-2001.

15926 Historically, mapped characters were remapped unless the **remap** edit option was not set, or the
15927 prefix of the mapped characters matched the mapping characters; for example, in the **map**:

15928 :map ab abcd

15929 the characters "ab" were used as is and were not remapped, but the characters "cd" were
15930 mapped if appropriate. This can cause infinite loops in the vi mapping mechanisms.
15931 IEEE Std 1003.1-2001 requires conformance to historical practice, and that such loops be
15932 interruptible.

15933 Text input maps had the same problems with expanding the *lhs* for the ex **map!** and **unmap!**
15934 command as did the ex **abbreviate** and **unabbreviate** commands. See the RATIONALE for the ex
15935 **abbreviate** command. IEEE Std 1003.1-2001 requires similar modification of some historical
15936 practice for the **map** and **unmap** commands, as described for the **abbreviate** and **unabbreviate**
15937 commands.

15938 Historically, **maps** that were subsets of other **maps** behaved differently depending on the order
15939 in which they were defined. For example:

15940 :map! ab short
15941 :map! abc long

15942 would always translate the characters "ab" to "short", regardless of how fast the characters
15943 "abc" were entered. If the entry order was reversed:

15944 :map! abc long
15945 :map! ab short

15946 the characters "ab" would cause the editor to pause, waiting for the completing 'c' character,
15947 and the characters might never be mapped to "short". For consistency and simplicity of
15948 specification, IEEE Std 1003.1-2001 requires that the shortest match be used at all times.

15949 The length of time the editor spends waiting for the characters to complete the *lhs* is unspecified
15950 because the timing capabilities of systems are often inexact and variable, and it may depend on
15951 other factors such as the speed of the connection. The time should be long enough for the user to
15952 be able to complete the sequence, but not long enough for the user to have to wait. Some
15953 implementations of vi have added a **keytime** option, which permits users to set the number of
15954 0.1 seconds the editor waits for the completing characters. Because mapped terminal function
15955 and cursor keys tend to start with an <ESC> character, and <ESC> is the key ending vi text input
15956 mode, **maps** starting with <ESC> characters are generally exempted from this timeout period,
15957 or, at least timed out differently.

15958 **Mark**

15959 Historically, users were able to set the "previous context" marks explicitly. In addition, the **ex**
15960 commands " and " and the **vi** commands ", ", and " all referred to the same mark. In addition,
15961 the previous context marks were not set if the command, with which the address setting the
15962 mark was associated, failed. IEEE Std 1003.1-2001 requires conformance to historical practice.
15963 Historically, if marked lines were deleted, the mark was also deleted, but would reappear if the
15964 change was undone. IEEE Std 1003.1-2001 requires conformance to historical practice.

15965 The description of the special events that set the ' and ' marks matches historical practice. For
15966 example, historically the command /a/,/b/ did not set the ' and ' marks, but the command
15967 /a/,/b/delete did.

15968 **Next**

15969 Historically, any **ex** command could be entered as a *+command* argument to the **next** command,
15970 although some (for example, **insert** and **append**) were known to confuse historical
15971 implementations. IEEE Std 1003.1-2001 requires that any command be permitted and that it
15972 behave as specified. The **next** command can accept more than one file, so usage such as:

15973 next `ls [abc] '

15974 is valid; it need not be valid for the **edit** or **read** commands, for example, because they expect
15975 only one filename.

15976 Historically, the **next** command behaved differently from the **:rewind** command in that it
15977 ignored the force flag if the **autowrite** flag was set. For consistency, IEEE Std 1003.1-2001 does
15978 not permit this behavior.

15979 Historically, the **next** command positioned the cursor as if the file had never been edited before,
15980 regardless. IEEE Std 1003.1-2001 does not permit this behavior, for consistency with the **edit**
15981 command.

15982 Implementations wanting to provide a counterpart to the **next** command that edited the
15983 previous file have used the command **prev[ious]**, which takes no *file* argument.

15984 IEEE Std 1003.1-2001 does not require this command.

15985 Open

15986 Historically, the **open** command would fail if the **open** edit option was not set.
15987 IEEE Std 1003.1-2001 does not mention the **open** edit option and does not require this behavior.
15988 Some historical implementations do not permit entering open mode from open or visual mode,
15989 only from ex mode. For consistency, IEEE Std 1003.1-2001 does not permit this behavior.

15990 Historically, entering open mode from the command line (that is, *vi +open*) resulted in
15991 anomalous behaviors; for example, the **ex** file and **set** commands, and the **vi** command
15992 <control>-G did not work. For consistency, IEEE Std 1003.1-2001 does not permit this behavior.

15993 Historically, the **open** command only permitted ' / ' characters to be used as the search pattern
15994 delimiter. For consistency, IEEE Std 1003.1-2001 requires that the search delimiters used by the **s**,
15995 **global**, and **v** commands be accepted as well.

15996 Preserve

15997 The **preserve** command does not historically cause the file to be considered unmodified for the
15998 purposes of future commands that may exit the editor. IEEE Std 1003.1-2001 requires
15999 conformance to historical practice.

16000 Historical documentation stated that mail was not sent to the user when **preserve** was executed;
16001 however, historical implementations did send mail in this case. IEEE Std 1003.1-2001 requires
16002 conformance to the historical implementations.

16003 Print

16004 The writing of NUL by the **print** command is not specified as a special case because the standard
16005 developers did not want to require **ex** to support NUL characters. Historically, characters were
16006 displayed using the ARPA standard mappings, which are as follows:

- 16007 1. Printable characters are left alone.
- 16008 2. Control characters less than \177 are represented as '^' followed by the character offset
16009 from the '@' character in the ASCII map; for example, \007 is represented as '^G'.
- 16010 3. \177 is represented as '^' followed by '?'.

16011 The display of characters having their eighth bit set was less standard. Existing implementations
16012 use hex (0x00), octal (\000), and a meta-bit display. (The latter displayed bytes that had their
16013 eighth bit set as the two characters "M-" followed by the seven-bit display as described above.)
16014 The latter probably has the best claim to historical practice because it was used for the -v option
16015 of 4 BSD and 4 BSD-derived versions of the **cat** utility since 1980.

16016 No specific display format is required by IEEE Std 1003.1-2001.

16017 Explicit dependence on the ASCII character set has been avoided where possible, hence the use
16018 of the phrase an "implementation-defined multi-character sequence" for the display of non-
16019 printable characters in preference to the historical usage of, for instance, "^I" for the <tab>. Implementations
16020 are encouraged to conform to historical practice in the absence of any strong
16021 reason to diverge.

16022 Historically, all **ex** commands beginning with the letter 'p' could be entered using capitalized
16023 versions of the commands; for example, **P[rint]**, **Pre[serve]**, and **Pu[t]** were all valid command
16024 names. IEEE Std 1003.1-2001 permits, but does not require, this historical practice because
16025 capital forms of the commands are used by some implementations for other purposes.

16026

Put

16027

Historically, an **ex put** command, executed from open or visual mode, was the same as the open or visual mode **P** command, if the buffer was named and was cut in character mode, and the same as the **p** command if the buffer was named and cut in line mode. If the unnamed buffer was the source of the text, the entire line from which the text was taken was usually **put**, and the buffer was handled as if in line mode, but it was possible to get extremely anomalous behavior. In addition, using the **Q** command to switch into **ex** mode, and then doing a **put** often resulted in errors as well, such as appending text that was unrelated to the (supposed) contents of the buffer. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit these behaviors. All **ex put** commands are required to operate in line mode, and the contents of the buffers are not altered by changing the mode of the editor.

16028

16029

16030

16031

16032

16033

16034

16035

16036

16037

Read

16038

16039

16040

16041

16042

Historically, an **ex read** command executed from open or visual mode, executed in an empty file, left an empty line as the first line of the file. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior. Historically, a **read** in open or visual mode from a program left the cursor at the last line read in, not the first. For consistency, IEEE Std 1003.1-2001 does not permit this behavior.

16043

16044

Historical implementations of **ex** were unable to undo **read** commands that read from the output of a program. For consistency, IEEE Std 1003.1-2001 does not permit this behavior.

16045

16046

16047

16048

16049

Historically, the **ex** and **vi** message after a successful **read** or **write** command specified “characters”, not “bytes”. IEEE Std 1003.1-2001 requires that the number of bytes be displayed, not the number of characters, because it may be difficult in multi-byte implementations to determine the number of characters read. Implementations are encouraged to clarify the message displayed to the user.

16050

16051

16052

16053

Historically, reads were not permitted on files other than type regular, except that FIFO files could be read (probably only because they did not exist when **ex** and **vi** were originally written). Because the historical **ex** evaluated **read!** and **read !** equivalently, there can be no optional way to force the read. IEEE Std 1003.1-2001 permits, but does not require, this behavior.

16054

Recover

16055

16056

16057

16058

Some historical implementations of the editor permitted users to recover the edit buffer contents from a previous edit session, and then exit without saving those contents (or explicitly discarding them). The intent of IEEE Std 1003.1-2001 in requiring that the edit buffer be treated as already modified is to prevent this user error.

16059

Rewind

16060

16061

16062

Historical implementations supported the **rewind** command when the user was editing the first file in the list; that is, the file that the **rewind** command would edit. IEEE Std 1003.1-2001 requires conformance to historical practice.

16063 **Substitute**

16064 Historically, *ex* accepted an **r** option to the **s** command. The effect of the **r** option was to use the
16065 last regular expression used in any command as the pattern, the same as the **~** command. The **r**
16066 option is not required by IEEE Std 1003.1-2001. Historically, the **c** and **g** options were toggled; for
16067 example, the command **:s/abc/def/** was the same as **s/abc/def/ccccgggg**. For simplicity of
16068 specification, IEEE Std 1003.1-2001 does not permit this behavior.

16069 The tilde command is often used to replace the last search RE. For example, in the sequence:

16070 **s/red/blue/**
16071 **/green**
16072 **~**

16073 the **~** command is equivalent to:

16074 **s/green/blue/**

16075 Historically, *ex* accepted all of the following forms:

16076 **s/abc/def/**
16077 **s/abc/def**
16078 **s/abc/**
16079 **s/abc**

16080 IEEE Std 1003.1-2001 requires conformance to this historical practice.

16081 The **s** command presumes that the **' '** character only occupies a single column in the display.
16082 Much of the *ex* and *vi* specification presumes that the **<space>** only occupies a single column in
16083 the display. There are no known character sets for which this is not true.

16084 Historically, the final column position for the substitute commands was based on previous
16085 column movements; a search for a pattern followed by a substitution would leave the column
16086 position unchanged, while a **0** command followed by a substitution would change the column
16087 position to the first non-<blank>. For consistency and simplicity of specification,
16088 IEEE Std 1003.1-2001 requires that the final column position always be set to the first non-
16089 <blank>.

16090 **Set**

16091 Historical implementations redisplayed all of the options for each occurrence of the **all** keyword.
16092 IEEE Std 1003.1-2001 permits, but does not require, this behavior.

16093 **Tag**

16094 No requirement is made as to where *ex* and *vi* shall look for the file referenced by the tag entry.
16095 Historical practice has been to look for the path found in the **tags** file, based on the current
16096 directory. A useful extension found in some implementations is to look based on the directory
16097 containing the **tags** file that held the entry, as well. No requirement is made as to which
16098 reference for the tag in the **tags** file is used. This is deliberate, in order to permit extensions such
16099 as multiple entries in a **tags** file for a tag.

16100 Because users often specify many different **tags** files, some of which need not be relevant or exist
16101 at any particular time, IEEE Std 1003.1-2001 requires that error messages about problem **tags**
16102 files be displayed only if the requested tag is not found, and then, only once for each time that
16103 the **tag** edit option is changed.

16104 The requirement that the current edit buffer be unmodified is only necessary if the file indicated
16105 by the **tag** entry is not the same as the current file (as defined by the current pathname).

16106 Historically, the file would be reloaded if the filename had changed, as well as if the filename
16107 was different from the current pathname. For consistency and simplicity of specification,
16108 IEEE Std 1003.1-2001 does not permit this behavior, requiring that the name be the only factor in
16109 the decision.

16110 Historically, *vi* only searched for tags in the current file from the current cursor to the end of the
16111 file, and therefore, if the **wrapscan** option was not set, tags occurring before the current cursor
16112 were not found. IEEE Std 1003.1-2001 considers this a bug, and implementations are required to
16113 search for the first occurrence in the file, regardless.

16114 **Undo**

16115 The **undo** description deliberately uses the word “modified”. The **undo** command is not
16116 intended to undo commands that replace the contents of the edit buffer, such as **edit**, **next**, **tag**,
16117 or **recover**.

16118 Cursor positioning after the **undo** command was inconsistent in the historical *vi*, sometimes
16119 attempting to restore the original cursor position (**global**, **undo**, and **v** commands), and
16120 sometimes, in the presence of maps, placing the cursor on the last line added or changed instead
16121 of the first. IEEE Std 1003.1-2001 requires a simplified behavior for consistency and simplicity of
16122 specification.

16123 **Version**

16124 The **version** command cannot be exactly specified since there is no widely-accepted definition of
16125 what the version information should contain. Implementations are encouraged to do something
16126 reasonably intelligent.

16127 **Write**

16128 Historically, the **ex** and *vi* message after a successful **read** or **write** command specified
16129 “characters”, not “bytes”. IEEE Std 1003.1-2001 requires that the number of bytes be displayed,
16130 not the number of characters because it may be difficult in multi-byte implementations to
16131 determine the number of characters written. Implementations are encouraged to clarify the
16132 message displayed to the user.

16133 Implementation-defined tests are permitted so that implementations can make additional
16134 checks; for example, for locks or file modification times.

16135 Historically, attempting to append to a nonexistent file caused an error. It has been left
16136 unspecified in IEEE Std 1003.1-2001 to permit implementations to let the **write** succeed, so that
16137 the append semantics are similar to those of the historical *csh*.

16138 Historical *vi* permitted empty edit buffers to be written. However, since the way *vi* got around
16139 dealing with “empty” files was to always have a line in the edit buffer, no matter what, it wrote
16140 them as files of a single, empty line. IEEE Std 1003.1-2001 does not permit this behavior.

16141 Historically, **ex** restored standard output and standard error to their values as of when **ex** was
16142 invoked, before writes to programs were performed. This could disturb the terminal
16143 configuration as well as be a security issue for some terminals. IEEE Std 1003.1-2001 does not
16144 permit this, requiring that the program output be captured and displayed as if by the **ex print**
16145 command.

- 16146 **Adjust Window**
- 16147 Historically, the line count was set to the value of the **scroll** option if the type character was
16148 end-of-file. This feature was broken on most historical implementations long ago, however, and
16149 is not documented anywhere. For this reason, IEEE Std 1003.1-2001 is resolutely silent.
- 16150 Historically, the **z** command was <blank>-sensitive and **z +** and **z -** did different things than **z+**
16151 and **z-** because the type could not be distinguished from a flag. (The commands **z .** and **z =**
16152 were historically invalid.) IEEE Std 1003.1-2001 requires conformance to this historical practice.
- 16153 Historically, the **z** command was further <blank>-sensitive in that the *count* could not be
16154 <blank>-delimited; for example, the commands **z= 5** and **z- 5** were also invalid. Because the
16155 *count* is not ambiguous with respect to either the type character or the flags, this is not permitted
16156 by IEEE Std 1003.1-2001.
- 16157 **Escape**
- 16158 Historically, **ex** filter commands only read the standard output of the commands, letting
16159 standard error appear on the terminal as usual. The **vi** utility, however, read both standard
16160 output and standard error. IEEE Std 1003.1-2001 requires the latter behavior for both **ex** and **vi**,
16161 for consistency.
- 16162 **Shift Left and Shift Right**
- 16163 Historically, it was possible to add shift characters to increase the effect of the command; for
16164 example, <<< outdented (or >>> indented) the lines 3 levels of indentation instead of the default
16165 1. IEEE Std 1003.1-2001 requires conformance to historical practice.
- 16166 **<control>-D**
- 16167 Historically, the <control>-D command erased the prompt, providing the user with an unbroken
16168 presentation of lines from the edit buffer. This is not required by IEEE Std 1003.1-2001;
16169 implementations are encouraged to provide it if possible. Historically, the <control>-D
16170 command took, and then ignored, a *count*. IEEE Std 1003.1-2001 does not permit this behavior.
- 16171 **Write Line Number**
- 16172 Historically, the **ex =** command, when executed in **ex** mode in an empty edit buffer, reported 0,
16173 and from open or visual mode, reported 1. For consistency and simplicity of specification,
16174 IEEE Std 1003.1-2001 does not permit this behavior.
- 16175 **Execute**
- 16176 Historically, **ex** did not correctly handle the inclusion of text input commands (that is, **append**,
16177 **insert**, and **change**) in executed buffers. IEEE Std 1003.1-2001 does not permit this exclusion for
16178 consistency.
- 16179 Historically, the logical contents of the buffer being executed did not change if the buffer itself
16180 were modified by the commands being executed; that is, buffer execution did not support self-
16181 modifying code. IEEE Std 1003.1-2001 requires conformance to historical practice.
- 16182 Historically, the **@** command took a range of lines, and the **@** buffer was executed once per line,
16183 with the current line ('.') set to each specified line. IEEE Std 1003.1-2001 requires conformance to
16184 historical practice.
- 16185 Some historical implementations did not notice if errors occurred during buffer execution. This,
16186 coupled with the ability to specify a range of lines for the **ex @** command, makes it trivial to
16187 cause them to drop **core**. IEEE Std 1003.1-2001 requires that implementations stop buffer

16188 execution if any error occurs, if the specified line doesn't exist, or if the contents of the edit buffer
16189 itself are replaced (for example, the buffer executes the **ex :edit** command).

16190 Regular Expressions in ex

16191 Historical practice is that the characters in the replacement part of the last **s** command—that is,
16192 those matched by entering a '`~`' in the regular expression—were not further expanded by the
16193 regular expression engine. So, if the characters contained the string "`a. ,`" they would match
16194 '`a`' followed by '`. ,`' and not '`a`' followed by any character. IEEE Std 1003.1-2001 requires
16195 conformance to historical practice.

16196 Edit Options in ex

16197 The following paragraphs describe the historical behavior of some edit options that were not, for
16198 whatever reason, included in IEEE Std 1003.1-2001. Implementations are strongly encouraged to
16199 only use these names if the functionality described here is fully supported.

- | | |
|------------------------|---|
| 16200 extended | The extended edit option has been used in some implementations of vi to provide
16201 extended regular expressions instead of basic regular expressions. This option was
16202 omitted from IEEE Std 1003.1-2001 because it is not widespread historical practice. |
| 16203 flash | The flash edit option historically caused the screen to flash instead of beeping on
16204 error. This option was omitted from IEEE Std 1003.1-2001 because it is not found in
16205 some historical implementations. |
| 16206 hardtabs | The hardtabs edit option historically defined the number of columns between
16207 hardware tab settings. This option was omitted from IEEE Std 1003.1-2001 because
16208 it was believed to no longer be generally useful. |
| 16209 modeline | The modeline (sometimes named modelines) edit option historically caused ex or
16210 vi to read the five first and last lines of the file for editor commands. This option is
16211 a security problem, and vendors are strongly encouraged to delete it from
16212 historical implementations. |
| 16213 open | The open edit option historically disallowed the ex open and visual commands.
16214 This edit option was omitted because these commands are required by
16215 IEEE Std 1003.1-2001. |
| 16216 optimize | The optimize edit option historically expedited text throughput by setting the
16217 terminal to not do automatic <carriage-return>s when printing more than one
16218 logical line of output. This option was omitted from IEEE Std 1003.1-2001 because
16219 it was intended for terminals without addressable cursors, which are rarely, if ever,
16220 still used. |
| 16221 ruler | The ruler edit option has been used in some implementations of vi to present a
16222 current row/column ruler for the user. This option was omitted from
16223 IEEE Std 1003.1-2001 because it is not widespread historical practice. |
| 16224 sourceany | The sourceany edit option historically caused ex or vi to source start-up files that
16225 were owned by users other than the user running the editor. This option is a
16226 security problem, and vendors are strongly encouraged to remove it from their
16227 implementations. |
| 16228 timeout | The timeout edit option historically enabled the (now standard) feature of only
16229 waiting for a short period before returning keys that could be part of a macro. This
16230 feature was omitted from IEEE Std 1003.1-2001 because its behavior is now
16231 standard, it is not widely useful, and it was rarely documented. |

16232	verbose	The verbose edit option has been used in some implementations of <i>vi</i> to cause <i>vi</i> to output error messages for common errors; for example, attempting to move the cursor past the beginning or end of the line instead of only alerting the screen. (The historical <i>vi</i> only alerted the terminal and presented no message for such errors. The historical editor option terse did not select when to present error messages, it only made existing error messages more or less verbose.) This option was omitted from IEEE Std 1003.1-2001 because it is not widespread historical practice; however, implementors are encouraged to use it if they wish to provide error messages for naive users.
16241	wraplen	The wraplen edit option has been used in some implementations of <i>vi</i> to specify an automatic margin measured from the left margin instead of from the right margin. This is useful when multiple screen sizes are being used to edit a single file. This option was omitted from IEEE Std 1003.1-2001 because it is not widespread historical practice; however, implementors are encouraged to use it if they add this functionality.
16247	autoindent, ai	Historically, the command 0a did not do any autoindentation, regardless of the current indentation of line 1. IEEE Std 1003.1-2001 requires that any indentation present in line 1 be used.
16250	autoprint, ap	Historically, the autoprint edit option was not completely consistent or based solely on modifications to the edit buffer. Exceptions were the read command (when reading from a file, but not from a filter), the append , change , insert , global , and v commands, all of which were not affected by autoprint , and the tag command, which was affected by autoprint . IEEE Std 1003.1-2001 requires conformance to historical practice.
16256		Historically, the autoprint option only applied to the last of multiple commands entered using vertical-bar delimiters; for example, delete <newline> was affected by autoprint , but delete version <newline> was not. IEEE Std 1003.1-2001 requires conformance to historical practice.
16260	autowrite, aw	Appending the '!' character to the ex next command to avoid performing an automatic write was not supported in historical implementations. IEEE Std 1003.1-2001 requires that the behavior match the other ex commands for consistency.
16264	ignorecase, ic	Historical implementations of case-insensitive matching (the ignorecase edit option) lead to counterintuitive situations when uppercase characters were used in range expressions. Historically, the process was as follows:
16268	1.	Take a line of text from the edit buffer.
16269	2.	Convert uppercase to lowercase in text line.
16270	3.	Convert uppercase to lowercase in regular expressions, except in character class specifications.
16272	4.	Match regular expressions against text.
16273		This would mean that, with ignorecase in effect, the text:

16274 The cat sat on the mat
16275 would be matched by
16276 /^the/
16277 but not by:
16278 /^ [A-Z]he/
16279 For consistency with other commands implementing regular expressions, IEEE Std 1003.1-2001
16280 does not permit this behavior.

16281 **paragraphs, para**

16282 The ISO POSIX-2:1993 standard made the default **paragraphs** and **sections** edit options
16283 implementation-defined, arguing they were historically oriented to the UNIX system *troff* text
16284 formatter, and a “portable user” could use the {, }, [[,]], (, and) commands in open or visual
16285 mode and have the cursor stop in unexpected places. IEEE Std 1003.1-2001 specifies their values
16286 in the POSIX locale because the unusual grouping (they only work when grouped into two
16287 characters at a time) means that they cannot be used for general-purpose movement, regardless.

16288 **readonly**

16289 Implementations are encouraged to provide the best possible information to the user as to the
16290 read-only status of the file, with the exception that they should not consider the current special
16291 privileges of the process. This provides users with a safety net because they must force the
16292 overwrite of read-only files, even when running with additional privileges.

16293 The **readonly** edit option specification largely conforms to historical practice. The only
16294 difference is that historical implementations did not notice that the user had set the **readonly**
16295 edit option in cases where the file was already marked read-only for some reason, and would
16296 therefore reinitialize the **readonly** edit option the next time the contents of the edit buffer were
16297 replaced. This behavior is disallowed by IEEE Std 1003.1-2001.

16298 **report**

16299 The requirement that lines copied to a buffer interact differently than deleted lines is historical
16300 practice. For example, if the **report** edit option is set to 3, deleting 3 lines will cause a report to be
16301 written, but 4 lines must be copied before a report is written.

16302 The requirement that the **ex global**, **v**, **open**, **undo**, and **visual** commands present reports based
16303 on the total number of lines added or deleted during the command execution, and that
16304 commands executed by the **global** and **v** commands not present reports, is historical practice.
16305 IEEE Std 1003.1-2001 extends historical practice by requiring that buffer execution be treated
16306 similarly. The reasons for this are two-fold. Historically, only the report by the last command
16307 executed from the buffer would be seen by the user, as each new report would overwrite the
16308 last. In addition, the standard developers believed that buffer execution had more in common
16309 with **global** and **v** commands than it did with other **ex** commands, and should behave similarly,
16310 for consistency and simplicity of specification.

16311 **showmatch, sm**

16312 The length of time the cursor spends on the matching character is unspecified because the
16313 timing capabilities of systems are often inexact and variable. The time should be long enough for
16314 the user to notice, but not long enough for the user to become annoyed. Some implementations
16315 of *vi* have added a **matchtime** option that permits users to set the number of 0.1 second intervals
16316 the cursor pauses on the matching character.

16317 **showmode**

16318 The **showmode** option has been used in some historical implementations of *ex* and *vi* to display
16319 the current editing mode when in open or visual mode. The editing modes have generally
16320 included “command” and “input”, and sometimes other modes such as “replace” and
16321 “change”. The string was usually displayed on the bottom line of the screen at the far right-hand
16322 corner. In addition, a preceding ‘*’ character often denoted whether the contents of the edit
16323 buffer had been modified. The latter display has sometimes been part of the **showmode** option,
16324 and sometimes based on another option. This option was not available in the 4 BSD historical
16325 implementation of *vi*, but was viewed as generally useful, particularly to novice users, and is
16326 required by IEEE Std 1003.1-2001.

16327 The **smd** shorthand for the **showmode** option was not present in all historical implementations
16328 of the editor. IEEE Std 1003.1-2001 requires it, for consistency.

16329 Not all historical implementations of the editor displayed a mode string for command mode,
16330 differentiating command mode from text input mode by the absence of a mode string.
16331 IEEE Std 1003.1-2001 permits this behavior for consistency with historical practice, but
16332 implementations are encouraged to provide a display string for both modes.

16333 **slowopen**

16334 Historically the **slowopen** option was automatically set if the terminal baud rate was less than
16335 1 200 baud, or if the baud rate was 1 200 baud and the **redraw** option was not set. The **slowopen**
16336 option had two effects. First, when inserting characters in the middle of a line, characters after
16337 the cursor would not be pushed ahead, but would appear to be overwritten. Second, when
16338 creating a new line of text, lines after the current line would not be scrolled down, but would
16339 appear to be overwritten. In both cases, ending text input mode would cause the screen to be
16340 refreshed to match the actual contents of the edit buffer. Finally, terminals that were sufficiently
16341 intelligent caused the editor to ignore the **slowopen** option. IEEE Std 1003.1-2001 permits most
16342 historical behavior, extending historical practice to require **slowopen** behaviors if the edit option
16343 is set by the user.

16344 **tags**

16345 The default path for tags files is left unspecified as implementations may have their own **tags**
16346 implementations that do not correspond to the historical ones. The default **tags** option value
16347 should probably at least include the file **./tags**.

- 16348 **term**
- 16349 Historical implementations of **ex** and **vi** ignored changes to the **term** edit option after the initial
16350 terminal information was loaded. This is permitted by IEEE Std 1003.1-2001; however,
16351 implementations are encouraged to permit the user to modify their terminal type at any time.
- 16352 **terse**
- 16353 Historically, the **terse** edit option optionally provided a shorter, less descriptive error message,
16354 for some error messages. This is permitted, but not required, by IEEE Std 1003.1-2001.
16355 Historically, most common visual mode errors (for example, trying to move the cursor past the
16356 end of a line) did not result in an error message, but simply alerted the terminal.
16357 Implementations wishing to provide messages for novice users are urged to do so based on the
16358 **edit** option **verbose**, and not **terse**.
- 16359 **window**
- 16360 In historical implementations, the default for the **window** edit option was based on the baud
16361 rate as follows:
- 16362 1. If the baud rate was less than 1 200, the **edit** option **w300** set the window value; for
16363 example, the line:

16364 `set w300=12`

16365 would set the window option to 12 if the baud rate was less than 1 200.
 - 16366 2. If the baud rate was equal to 1 200, the **edit** option **w1200** set the window value.
 - 16367 3. If the baud rate was greater than 1 200, the **edit** option **w9600** set the window value.
- 16368 The **w300**, **w1200**, and **w9600** options do not appear in IEEE Std 1003.1-2001 because of their
16369 dependence on specific baud rates.
- 16370 In historical implementations, the size of the window displayed by various commands was
16371 related to, but not necessarily the same as, the **window** edit option. For example, the size of the
16372 window was set by the **ex** command **visual 10**, but it did not change the value of the **window**
16373 edit option. However, changing the value of the **window** edit option did change the number of
16374 lines that were displayed when the screen was repainted. IEEE Std 1003.1-2001 does not permit
16375 this behavior in the interests of consistency and simplicity of specification, and requires that all
16376 commands that change the number of lines that are displayed do it by setting the value of the
16377 **window** edit option.
- 16378 **wrapmargin, wm**
- 16379 Historically, the **wrapmargin** option did not affect maps inserting characters that also had
16380 associated *counts*; for example `:map K 5aABC DEF`. Unfortunately, there are widely used
16381 maps that depend on this behavior. For consistency and simplicity of specification,
16382 IEEE Std 1003.1-2001 does not permit this behavior.
- 16383 Historically, **wrapmargin** was calculated using the column display width of all characters on the
16384 screen. For example, an implementation using "`^I`" to represent <tab>s when the **list** edit
16385 option was set, where '`^`' and '`I`' each took up a single column on the screen, would calculate
16386 the **wrapmargin** based on a value of 2 for each <tab>. The **number** edit option similarly
16387 changed the effective length of the line as well. IEEE Std 1003.1-2001 requires conformance to
16388 historical practice.

16389 FUTURE DIRECTIONS

16390 None.

16391 SEE ALSO

16392 Section 2.9.1.1 (on page 48), *ctags*, *ed*, *sed*, *sh*, *stty*, *vi*, the System Interfaces volume of
16393 IEEE Std 1003.1-2001, *access()*

16394 CHANGE HISTORY

16395 First released in Issue 2.

16396 Issue 5

16397 The FUTURE DIRECTIONS section is added.

16398 Issue 6

16399 This utility is marked as part of the User Portability Utilities option.

16400 The obsolescent SYNOPSIS is removed, removing the *+command* and *-* options.

16401 The following new requirements on POSIX implementations derive from alignment with the
16402 Single UNIX Specification:

- In the **map** command description, the sequence *#digit* is added.

- The **directory**, **edcompatible**, **redraw**, and **slowopen** edit options are added.

16403 The **ex** utility is extensively changed for alignment with the IEEE P1003.2b draft standard. This
16404 includes changes as a result of the IEEE PASC Interpretations 1003.2 #31, #38, #49, #50, #51, #52,
16405 #55, #56, #57, #61, #62, #63, #64, #65, and #78.

16406 The **-l** option is removed.

16409 NAME

16410 expand — convert tabs to spaces

16411 SYNOPSIS

16412 UP `expand [-t tablist][file ...]`

16413

16414 DESCRIPTION

16415 The *expand* utility shall write files or the standard input to the standard output with <tab>s replaced with one or more <space>s needed to pad to the next tab stop. Any <backspace>s shall be copied to the output and cause the column position count for tab stop calculations to be decremented; the column position count shall not be decremented below zero.

16419 OPTIONS

16420 The *expand* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

16422 The following option shall be supported:

16423 **-t tablist** Specify the tab stops. The application shall ensure that the argument *tablist* consists of either a single positive decimal integer or a list of tabstops. If a single number is given, tabs shall be set that number of column positions apart instead of the default 8.

16427 If a list of tabstops is given, the application shall ensure that it consists of a list of two or more positive decimal integers, separated by <blank>s or commas, in ascending order. The tabs shall be set at those specific column positions. Each tab stop *N* shall be an integer value greater than zero, and the list is in strictly ascending order. This is taken to mean that, from the start of a line of output, tabbing to position *N* shall cause the next character output to be in the (*N*+1)th column position on that line.

16434 In the event of *expand* having to process a <tab> at a position beyond the last of those specified in a multiple tab-stop list, the <tab> shall be replaced by a single <space> in the output.

16437 OPERANDS

16438 The following operand shall be supported:

16439 *file* The pathname of a text file to be used as input.

16440 STDIN

16441 See the INPUT FILES section.

16442 INPUT FILES

16443 Input files shall be text files.

16444 ENVIRONMENT VARIABLES

16445 The following environment variables shall affect the execution of *expand*:

16446 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

16450 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

16452 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in

16454 arguments and input files), the processing of <tab>s and <space>s, and for the
16455 determination of the width in column positions each character would occupy on
16456 an output device.

16457 ***LC_MESSAGES***

16458 Determine the locale that should be used to affect the format and contents of
16459 diagnostic messages written to standard error.

16460 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of ***LC_MESSAGES***.

16461 **ASYNCHRONOUS EVENTS**

16462 Default.

16463 **STDOUT**

16464 The standard output shall be equivalent to the input files with <tab>s converted into the
16465 appropriate number of <space>s.

16466 **STDERR**

16467 The standard error shall be used only for diagnostic messages.

16468 **OUTPUT FILES**

16469 None.

16470 **EXTENDED DESCRIPTION**

16471 None.

16472 **EXIT STATUS**

16473 The following exit values shall be returned:

16474 0 Successful completion

16475 >0 An error occurred.

16476 **CONSEQUENCES OF ERRORS**

16477 The *expand* utility shall terminate with an error message and non-zero exit status upon
16478 encountering difficulties accessing one of the *file* operands.

16479 **APPLICATION USAGE**

16480 None.

16481 **EXAMPLES**

16482 None.

16483 **RATIONALE**

16484 The *expand* utility is useful for preprocessing text files (before sorting, looking at specific
16485 columns, and so on) that contain <tab>s.

16486 See the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.103, Column Position.

16487 The *tablist* option-argument consists of integers in ascending order. Utility Syntax Guideline 8
16488 mandates that *expand* shall accept the integers (within the single argument) separated using
16489 either commas or <blank>s.

16490 **FUTURE DIRECTIONS**

16491 None.

16492 **SEE ALSO**

16493 *tabs*, *unexpand*

16494 CHANGE HISTORY

16495 First released in Issue 4.

16496 Issue 6

16497 This utility is marked as part of the User Portability Utilities option.

16498 The APPLICATION USAGE section is added.

16499 The obsolescent SYNOPSIS is removed.

16500 The *LC_CTYPE* environment variable description is updated to align with the IEEE P1003.2b draft standard.

16502 The normative text is reworded to avoid use of the term “must” for application requirements.

16503 **NAME**16504 *expr* — evaluate arguments as an expression16505 **SYNOPSIS**16506 *expr operand*16507 **DESCRIPTION**16508 The *expr* utility shall evaluate an expression and write the result to standard output.16509 **OPTIONS**

16510 None.

16511 **OPERANDS**16512 The single expression evaluated by *expr* shall be formed from the operands, as described in the
16513 EXTENDED DESCRIPTION section. The application shall ensure that each of the expression
16514 operator symbols:

16515 () | & = > >= < <= != + - * / % :

16516 and the symbols *integer* and *string* in the table are provided as separate arguments to *expr*.16517 **STDIN**

16518 Not used.

16519 **INPUT FILES**

16520 None.

16521 **ENVIRONMENT VARIABLES**16522 The following environment variables shall affect the execution of *expr*:16523 ***LANG*** Provide a default value for the internationalization variables that are unset or null.
16524 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
16525 Internationalization Variables for the precedence of internationalization variables
16526 used to determine the values of locale categories.)16527 ***LC_ALL*** If set to a non-empty string value, override the values of all the other
16528 internationalization variables.16529 ***LC_COLLATE***16530 Determine the locale for the behavior of ranges, equivalence classes, and multi-
16531 character collating elements within regular expressions and by the string
16532 comparison operators.16533 ***LC_CTYPE*** Determine the locale for the interpretation of sequences of bytes of text data as
16534 characters (for example, single-byte as opposed to multi-byte characters in
16535 arguments) and the behavior of character classes within regular expressions.16536 ***LC_MESSAGES***16537 Determine the locale that should be used to affect the format and contents of
16538 diagnostic messages written to standard error.16539 **XSI** ***NLSPATH*** Determine the location of message catalogs for the processing of *LC_MESSAGES*.16540 **ASYNCHRONOUS EVENTS**

16541 Default.

16542 **STDOUT**16543 The *expr* utility shall evaluate the expression and write the result, followed by a <newline>, to
16544 standard output.

16545 **STDERR**

16546 The standard error shall be used only for diagnostic messages.

16547 **OUTPUT FILES**

16548 None.

16549 **EXTENDED DESCRIPTION**

16550 The formation of the expression to be evaluated is shown in the following table. The symbols
 16551 *expr*, *expr1*, and *expr2* represent expressions formed from *integer* and *string* symbols and the
 16552 expression operator symbols (all separate arguments) by recursive application of the constructs
 16553 described in the table. The expressions are listed in order of increasing precedence, with equal-
 16554 precedence operators grouped between horizontal lines. All of the operators shall be left-
 16555 associative.

Expression	Description
<i>expr1</i> <i>expr2</i>	Returns the evaluation of <i>expr1</i> if it is neither null nor zero; otherwise, returns the evaluation of <i>expr2</i> if it is not null; otherwise, zero.
<i>expr1</i> & <i>expr2</i>	Returns the evaluation of <i>expr1</i> if neither expression evaluates to null or zero; otherwise, returns zero.
<i>expr1</i> = <i>expr2</i> <i>expr1</i> > <i>expr2</i> <i>expr1</i> >= <i>expr2</i> <i>expr1</i> < <i>expr2</i> <i>expr1</i> <= <i>expr2</i> <i>expr1</i> != <i>expr2</i>	Returns the result of a decimal integer comparison if both arguments are integers; otherwise, returns the result of a string comparison using the locale-specific collation sequence. The result of each comparison is 1 if the specified relationship is true, or 0 if the relationship is false. Equal. Greater than. Greater than or equal. Less than. Less than or equal. Not equal.
<i>expr1</i> + <i>expr2</i> <i>expr1</i> - <i>expr2</i>	Addition of decimal integer-valued arguments. Subtraction of decimal integer-valued arguments.
<i>expr1</i> * <i>expr2</i> <i>expr1</i> / <i>expr2</i> <i>expr1</i> % <i>expr2</i>	Multiplication of decimal integer-valued arguments. Integer division of decimal integer-valued arguments, producing an integer result. Remainder of integer division of decimal integer-valued arguments.
<i>expr1</i> : <i>expr2</i>	Matching expression; see below.
(<i>expr</i>)	Grouping symbols. Any expression can be placed within parentheses. Parentheses can be nested to a depth of {EXPR_NEST_MAX}.
<i>integer</i> <i>string</i>	An argument consisting only of an (optional) unary minus followed by digits. A string argument; see below.

16587

Matching Expression

16588

The ': ' matching operator shall compare the string resulting from the evaluation of *expr1* with the regular expression pattern resulting from the evaluation of *expr2*. Regular expression syntax shall be that defined in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions, except that all patterns are anchored to the beginning of the string (that is, only sequences starting at the first character of a string are matched by the regular expression) and, therefore, it is unspecified whether '^' is a special character in that context. Usually, the matching operator shall return a string representing the number of characters matched ('0' on failure). Alternatively, if the pattern contains at least one regular expression subexpression "[\(\ .\)]", the string corresponding to "\1" shall be returned.

16589

16590

16591

16592

16593

16594

16595

16596

16597

String Operand

16598

16599

A string argument is an argument that cannot be identified as an *integer* argument or as one of the expression operator symbols shown in the OPERANDS section.

16600

The use of string arguments **length**, **substr**, **index**, or **match** produces unspecified results.

16601

EXIT STATUS

16602

The following exit values shall be returned:

16603

0 The *expression* evaluates to neither null nor zero.

16604

1 The *expression* evaluates to null or zero.

16605

2 Invalid *expression*.

16606

>2 An error occurred.

16607

CONSEQUENCES OF ERRORS

16608

Default.

16609

APPLICATION USAGE

16610

After argument processing by the shell, *expr* is not required to be able to tell the difference between an operator and an operand except by the value. If "\$a" is '=' , the command:

16612

```
expr $a = '='
```

16613

looks like:

16614

```
expr = = =
```

16615

as the arguments are passed to *expr* (and they all may be taken as the '=' operator). The following works reliably:

16617

```
expr X$a = X=
```

16618

Also note that this volume of IEEE Std 1003.1-2001 permits implementations to extend utilities. The *expr* utility permits the integer arguments to be preceded with a unary minus. This means that an integer argument could look like an option. Therefore, the conforming application must employ the "--" construct of Guideline 10 of the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines to protect its operands if there is any chance the first operand might be a negative integer (or any string with a leading minus).

16624

EXAMPLES

16625

The *expr* utility has a rather difficult syntax:

16626

- Many of the operators are also shell control operators or reserved words, so they have to be escaped on the command line.

- 16628 • Each part of the expression is composed of separate arguments, so liberal usage of <blank>s
 16629 is required. For example:

Invalid	Valid
<code>expr 1+2</code>	<code>expr 1 + 2</code>
<code>expr "1 + 2"</code>	<code>expr 1 + 2</code>
<code>expr 1 + (2 * 3)</code>	<code>expr 1 + \((2 \(*\ 3 \))\)</code>

16635 In many cases, the arithmetic and string features provided as part of the shell command
 16636 language are easier to use than their equivalents in *expr*. Newly written scripts should avoid
 16637 *expr* in favor of the new features within the shell; see Section 2.5 (on page 33) and Section 2.6.4
 16638 (on page 41).

16639 The following command:

16640 `a=$(expr $a + 1)`

16641 adds 1 to the variable *a*.

16642 The following command, for "\$a" equal to either /usr/abc/file or just file:

16643 `expr $a : '.* /\(.*\)' \| $a`

16644 returns the last segment of a pathname (that is, *file*). Applications should avoid the character
 16645 '/' used alone as an argument; *expr* may interpret it as the division operator.

16646 The following command:

16647 `expr "//$a" : '.* /\(.*\)'`

16648 is a better representation of the previous example. The addition of the "://" characters
 16649 eliminates any ambiguity about the division operator and simplifies the whole expression. Also
 16650 note that pathnames may contain characters contained in the *IFS* variable and should be quoted
 16651 to avoid having "\$a" expand into multiple arguments.

16652 The following command:

16653 `expr "$VAR" : '.*'`

16654 returns the number of characters in *VAR*.

16655 RATIONALE

16656 In an early proposal, EREs were used in the matching expression syntax. This was changed to
 16657 BREs to avoid breaking historical applications.

16658 The use of a leading circumflex in the BRE is unspecified because many historical
 16659 implementations have treated it as a special character, despite their system documentation. For
 16660 example:

16661 `expr foo : ^foo expr ^foo : ^foo`

16662 return 3 and 0, respectively, on those systems; their documentation would imply the reverse.
 16663 Thus, the anchoring condition is left unspecified to avoid breaking historical scripts relying on
 16664 this undocumented feature.

16665 FUTURE DIRECTIONS

16666 None.

16667 SEE ALSO

16668 Section 2.5 (on page 33), Section 2.6.4 (on page 41)

16669 CHANGE HISTORY

16670 First released in Issue 2.

16671 Issue 5

16672 The FUTURE DIRECTIONS section is added.

16673 Issue 6

16674 The *expr* utility is aligned with the IEEE P1003.2b draft standard, to include resolution of IEEE
16675 PASC Interpretation 1003.2 #104.

16676 The normative text is reworded to avoid use of the term “must” for application requirements.

16677 NAME

16678 **false** — return false value

16679 SYNOPSIS

16680 **false**

16681 DESCRIPTION

16682 The *false* utility shall return with a non-zero exit code.

16683 OPTIONS

16684 None.

16685 OPERANDS

16686 None.

16687 STDIN

16688 Not used.

16689 INPUT FILES

16690 None.

16691 ENVIRONMENT VARIABLES

16692 None.

16693 ASYNCHRONOUS EVENTS

16694 Default.

16695 STDOUT

16696 Not used.

16697 STDERR

16698 None.

16699 OUTPUT FILES

16700 None.

16701 EXTENDED DESCRIPTION

16702 None.

16703 EXIT STATUS

16704 The *false* utility shall always exit with a value other than zero.

16705 CONSEQUENCES OF ERRORS

16706 Default.

16707 APPLICATION USAGE

16708 None.

16709 EXAMPLES

16710 None.

16711 RATIONALE

16712 None.

16713 FUTURE DIRECTIONS

16714 None.

16715 SEE ALSO

16716 *true*

16717 CHANGE HISTORY

16718 First released in Issue 2.

16719 NAME

16720 fc — process the command history list

16721 SYNOPSIS

16722 UP fc [-r][-e editor] [first[last]]

16723 fc -l[-nr] [first[last]]

16724 fc -s[old=new][first]

16725

16726 DESCRIPTION

16727 The *fc* utility shall list, or shall edit and re-execute, commands previously entered to an
16728 interactive *sh*.

16729 The command history list shall reference commands by number. The first number in the list is
16730 selected arbitrarily. The relationship of a number to its command shall not change except when
16731 the user logs in and no other process is accessing the list, at which time the system may reset the
16732 numbering to start the oldest retained command at another number (usually 1). When the
16733 number reaches an implementation-defined upper limit, which shall be no smaller than the
16734 value in *HISTSIZE* or 32 767 (whichever is greater), the shell may wrap the numbers, starting the
16735 next command with a lower number (usually 1). However, despite this optional wrapping of
16736 numbers, *fc* shall maintain the time-ordering sequence of the commands. For example, if four
16737 commands in sequence are given the numbers 32 766, 32 767, 1 (wrapped), and 2 as they are
16738 executed, command 32 767 is considered the command previous to 1, even though its number is
16739 higher.

16740 When commands are edited (when the **-l** option is not specified), the resulting lines shall be
16741 entered at the end of the history list and then re-executed by *sh*. The *fc* command that caused the
16742 editing shall not be entered into the history list. If the editor returns a non-zero exit status, this
16743 shall suppress the entry into the history list and the command re-execution. Any command line
16744 variable assignments or redirection operators used with *fc* shall affect both the *fc* command itself
16745 as well as the command that results; for example:

16746 fc -s -- -l 2>/dev/null

16747 reinvoques the previous command, suppressing standard error for both *fc* and the previous
16748 command.

16749 OPTIONS

16750 The *fc* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
16751 Utility Syntax Guidelines.

16752 The following options shall be supported:

16753 **-e editor** Use the editor named by *editor* to edit the commands. The *editor* string is a utility
16754 name, subject to search via the *PATH* variable (see the Base Definitions volume of
16755 IEEE Std 1003.1-2001, Chapter 8, Environment Variables). The value in the *FCEDIT*
16756 variable shall be used as a default when **-e** is not specified. If *FCEDIT* is null or
16757 unset, *ed* shall be used as the editor.

16758 **-l** (The letter ell.) List the commands rather than invoking an editor on them. The
16759 commands shall be written in the sequence indicated by the *first* and *last* operands,
16760 as affected by **-r**, with each command preceded by the command number.

16761 **-n** Suppress command numbers when listing with **-l**.

16762 **-r** Reverse the order of the commands listed (with **-l**) or edited (with neither **-l** nor
16763 **-s**).

16764 **-s** Re-execute the command without invoking an editor.

16765 **OPERANDS**

16766 The following operands shall be supported:

16767 *first, last* Select the commands to list or edit. The number of previous commands that can be accessed shall be determined by the value of the *HISTSIZE* variable. The value of *first* or *last* or both shall be one of the following:

16770 **[+]number** A positive number representing a command number; command numbers can be displayed with the **-l** option.

16772 **-number** A negative decimal number representing the command that was executed *number* of commands previously. For example, **-1** is the immediately previous command.

16775 **string** A string indicating the most recently entered command that begins with that string. If the *old=new* operand is not also specified with **-s**, the string form of the *first* operand cannot contain an embedded equal sign.

16779 When the synopsis form with **-s** is used:

- If *first* is omitted, the previous command shall be used.

16781 For the synopsis forms without **-s**:

16782 • If *last* is omitted, *last* shall default to the previous command when **-l** is specified; otherwise, it shall default to *first*.

16784 • If *first* and *last* are both omitted, the previous 16 commands shall be listed or the previous single command shall be edited (based on the **-l** option).

16786 • If *first* and *last* are both present, all of the commands from *first* to *last* shall be edited (without **-l**) or listed (with **-l**). Editing multiple commands shall be accomplished by presenting to the editor all of the commands at one time, each command starting on a new line. If *first* represents a newer command than *last*, the commands shall be listed or edited in reverse sequence, equivalent to using **-r**. For example, the following commands on the first line are equivalent to the corresponding commands on the second:

16793 **fc -r 10 20 fc 30 40**
 16794 **fc 20 10 fc -r 40 30**

16795 • When a range of commands is used, it shall not be an error to specify *first* or *last* values that are not in the history list; *fc* shall substitute the value representing the oldest or newest command in the list, as appropriate. For example, if there are only ten commands in the history list, numbered 1 to 10:

16799 **fc -l**
 16800 **fc 1 99**

16801 shall list and edit, respectively, all ten commands.

16802 **old=new** Replace the first occurrence of string *old* in the commands to be re-executed by the string *new*.

16804 **STDIN**

16805 Not used.

16806 **INPUT FILES**

16807 None.

16808 **ENVIRONMENT VARIABLES**16809 The following environment variables shall affect the execution of *fc*:

16810 **FCEDIT** This variable, when expanded by the shell, shall determine the default value for
16811 the **-e editor** option's *editor* option-argument. If *FCEDIT* is null or unset, *ed* shall be
16812 used as the editor.

16813 **HISTFILE** Determine a pathname naming a command history file. If the *HISTFILE* variable is
16814 not set, the shell may attempt to access or create a file **.sh_history** in the directory
16815 referred to by the *HOME* environment variable. If the shell cannot obtain both read
16816 and write access to, or create, the history file, it shall use an unspecified
16817 mechanism that allows the history to operate properly. (References to history
16818 "file" in this section shall be understood to mean this unspecified mechanism in
16819 such cases.) An implementation may choose to access this variable only when
16820 initializing the history file; this initialization shall occur when *fc* or *sh* first attempt
16821 to retrieve entries from, or add entries to, the file, as the result of commands issued
16822 by the user, the file named by the *ENV* variable, or implementation-defined system
16823 start-up files. In some historical shells, the history file is initialized just after the
16824 *ENV* file has been processed. Therefore, it is implementation-defined whether
16825 changes made to *HISTFILE* after the history file has been initialized are effective.
16826 Implementations may choose to disable the history list mechanism for users with
16827 appropriate privileges who do not set *HISTFILE*; the specific circumstances under
16828 which this occurs are implementation-defined. If more than one instance of the
16829 shell is using the same history file, it is unspecified how updates to the history file
16830 from those shells interact. As entries are deleted from the history file, they shall be
16831 deleted oldest first. It is unspecified when history file entries are physically
16832 removed from the history file.

16833 **HISTSIZE** Determine a decimal number representing the limit to the number of previous
16834 commands that are accessible. If this variable is unset, an unspecified default
16835 greater than or equal to 128 shall be used. The maximum number of commands in
16836 the history list is unspecified, but shall be at least 128. An implementation may
16837 choose to access this variable only when initializing the history file, as described
16838 under *HISTFILE*. Therefore, it is unspecified whether changes made to *HISTSIZE*
16839 after the history file has been initialized are effective.

16840 **LANG** Provide a default value for the internationalization variables that are unset or null.
16841 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
16842 Internationalization Variables for the precedence of internationalization variables
16843 used to determine the values of locale categories.)

16844 **LC_ALL** If set to a non-empty string value, override the values of all the other
16845 internationalization variables.

16846 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
16847 characters (for example, single-byte as opposed to multi-byte characters in
16848 arguments and input files).

16849 **LC_MESSAGES** Determine the locale that should be used to affect the format and contents of
16850 diagnostic messages written to standard error.
16851

16852 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

16853 ASYNCHRONOUS EVENTS

16854 Default.

16855 STDOUT

16856 When the **-l** option is used to list commands, the format of each command in the list shall be as
16857 follows:

16858 "%d\t%s\n", <line number>, <command>

16859 If both the **-l** and **-n** options are specified, the format of each command shall be:

16860 "\t%s\n", <command>

16861 If the <command> consists of more than one line, the lines after the first shall be displayed as:

16862 "\t%s\n", <continued-command>

16863 STDERR

16864 The standard error shall be used only for diagnostic messages.

16865 OUTPUT FILES

16866 None.

16867 EXTENDED DESCRIPTION

16868 None.

16869 EXIT STATUS

16870 The following exit values shall be returned:

16871 0 Successful completion of the listing.

16872 >0 An error occurred.

16873 Otherwise, the exit status shall be that of the commands executed by *fc*.

16874 CONSEQUENCES OF ERRORS

16875 Default.

16876 APPLICATION USAGE

16877 Since editors sometimes use file descriptors as integral parts of their editing, redirecting their file
16878 descriptors as part of the *fc* command can produce unexpected results. For example, if *vi* is the
16879 *FCEDIT* editor, the command:

16880 fc -s | more

16881 does not work correctly on many systems.

16882 Users on windowing systems may want to have separate history files for each window by
16883 setting *HISTFILE* as follows:

16884 HISTFILE=\$HOME/.sh_hist\$\$

16885 EXAMPLES

16886 None.

16887 RATIONALE

16888 This utility is based on the *fc* built-in of the KornShell.

16889 An early proposal specified the **-e** option as **[**-e** editor [*old= new*]]**, which is not historical
16890 practice. Historical practice in *fc* of either **[**-e** editor]** or **[**-e** [*old= new*]]** is acceptable, but not
16891 both together. To clarify this, a new option **-s** was introduced replacing the **[**-e** -]**. This resolves
16892 the conflict and makes *fc* conform to the Utility Syntax Guidelines.

16893 **HISTFILE** Some implementations of the KornShell check for the superuser and do not create
16894 a history file unless *HISTFILE* is set. This is done primarily to avoid creating
16895 unlinked files in the root file system when logging in during single-user mode.
16896 *HISTFILE* must be set for the superuser to have history.

16897 **HISTSIZE** Needed to limit the size of history files. It is the intent of the standard developers
16898 that when two shells share the same history file, commands that are entered in one
16899 shell shall be accessible by the other shell. Because of the difficulties of
16900 synchronization over a network, the exact nature of the interaction is unspecified.

16901 The initialization process for the history file can be dependent on the system start-up files, in
16902 that they may contain commands that effectively preempt the settings the user has for *HISTFILE*
16903 and *HISTSIZE*. For example, function definition commands are recorded in the history file. If the
16904 system administrator includes function definitions in some system start-up file called before the
16905 *ENV* file, the history file is initialized before the user can influence its characteristics. In some
16906 historical shells, the history file is initialized just after the *ENV* file has been processed. Because
16907 of these situations, the text requires the initialization process to be implementation-defined.

16908 Consideration was given to omitting the *fc* utility in favor of the command line editing feature in
16909 *sh*. For example, in *vi* editing mode, typing "<ESC> v" is equivalent to:

16910 EDITOR=vi fc

16911 However, the *fc* utility allows the user the flexibility to edit multiple commands simultaneously
16912 (such as *fc* 10 20) and to use editors other than those supported by *sh* for command line editing.

16913 In the KornShell, the alias **r** ('re-do') is preset to *fc -e* – (equivalent to the POSIX *fc -s*). This is
16914 probably an easier command name to remember than *fc* ("fix command"), but it does not meet
16915 the Utility Syntax Guidelines. Renaming *fc* to *hist* or *redo* was considered, but since this
16916 description closely matches historical KornShell practice already, such a renaming was seen as
16917 gratuitous. Users are free to create aliases whenever odd historical names such as *fc*, *awk*, *cat*,
16918 *grep*, or *yacc* are standardized by POSIX.

16919 Command numbers have no ordering effects; they are like serial numbers. The **-r** option and
16920 **-number** operand address the sequence of command execution, regardless of serial numbers. So,
16921 for example, if the command number wrapped back to 1 at some arbitrary point, there would be
16922 no ambiguity associated with traversing the wrap point. For example, if the command history
16923 were:

16924 32766: echo 1
16925 32767: echo 2
16926 1: echo 3

16927 the number -2 refers to command 32767 because it is the second previous command, regardless
16928 of serial number.

16929 **FUTURE DIRECTIONS**

16930 None.

16931 **SEE ALSO**

16932 *sh*

16933 **CHANGE HISTORY**

16934 First released in Issue 4.

16935 Issue 5

16936 The FUTURE DIRECTIONS section is added.

16937 Issue 6

16938 This utility is marked as part of the User Portability Utilities option.

16939 In the ENVIRONMENT VARIABLES section, the text “user’s home directory” is updated to
16940 “directory referred to by the *HOME* environment variable”.

16941 NAME

16942 *fg* — run jobs in the foreground

16943 SYNOPSIS

16944 UP **fg** [*job_id*]

16945

16946 DESCRIPTION

16947 If job control is enabled (see the description of *set -m*), the *fg* utility shall move a background job
16948 from the current environment (see Section 2.12 (on page 61)) into the foreground.

16949 Using *fg* to place a job into the foreground shall remove its process ID from the list of those
16950 “known in the current shell execution environment”; see Section 2.9.3.1 (on page 50).

16951 OPTIONS

16952 None.

16953 OPERANDS

16954 The following operand shall be supported:

16955 *job_id* Specify the job to be run as a foreground job. If no *job_id* operand is given, the
16956 *job_id* for the job that was most recently suspended, placed in the background, or
16957 run as a background job shall be used. The format of *job_id* is described in the Base
16958 Definitions volume of IEEE Std 1003.1-2001, Section 3.203, Job Control Job ID.

16959 STDIN

16960 Not used.

16961 INPUT FILES

16962 None.

16963 ENVIRONMENT VARIABLES

16964 The following environment variables shall affect the execution of *fg*:

16965 *LANG* Provide a default value for the internationalization variables that are unset or null.
16966 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
16967 Internationalization Variables for the precedence of internationalization variables
16968 used to determine the values of locale categories.)

16969 *LC_ALL* If set to a non-empty string value, override the values of all the other
16970 internationalization variables.

16971 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
16972 characters (for example, single-byte as opposed to multi-byte characters in
16973 arguments).

16974 *LC_MESSAGES*

16975 Determine the locale that should be used to affect the format and contents of
16976 diagnostic messages written to standard error.

16977 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

16978 ASYNCHRONOUS EVENTS

16979 Default.

16980 STDOUT

16981 The *fg* utility shall write the command line of the job to standard output in the following format:

16982 "%s\n", <command>

16983 STDERR

16984 The standard error shall be used only for diagnostic messages.

16985 OUTPUT FILES

16986 None.

16987 EXTENDED DESCRIPTION

16988 None.

16989 EXIT STATUS

16990 The following exit values shall be returned:

16991 0 Successful completion.

16992 >0 An error occurred.

16993 CONSEQUENCES OF ERRORS

16994 If job control is disabled, the *fg* utility shall exit with an error and no job shall be placed in the foreground.

16996 APPLICATION USAGE

16997 The *fg* utility does not work as expected when it is operating in its own utility execution environment because that environment has no applicable jobs to manipulate. See the APPLICATION USAGE section for *bg*. For this reason, *fg* is generally implemented as a shell regular built-in.

17001 EXAMPLES

17002 None.

17003 RATIONALE

17004 The extensions to the shell specified in this volume of IEEE Std 1003.1-2001 have mostly been based on features provided by the KornShell. The job control features provided by *bg*, *fg*, and *jobs* are also based on the KornShell. The standard developers examined the characteristics of the C shell versions of these utilities and found that differences exist. Despite widespread use of the C shell, the KornShell versions were selected for this volume of IEEE Std 1003.1-2001 to maintain a degree of uniformity with the rest of the KornShell features selected (such as the very popular command line editing features).

17011 FUTURE DIRECTIONS

17012 None.

17013 SEE ALSO

17014 Section 2.9.3.1 (on page 50), Section 2.12 (on page 61), *bg*, *kill*, *jobs*, *wait*

17015 CHANGE HISTORY

17016 First released in Issue 4.

17017 Issue 6

17018 This utility is marked as part of the User Portability Utilities option.

17019 The APPLICATION USAGE section is added.

17020 The JC marking is removed from the SYNOPSIS since job control is mandatory in this issue.

17021 NAME

17022 file — determine file type

17023 SYNOPSIS

17024 UP file [-dhi][-M file][-m file] file ...

17025

17026 DESCRIPTION

17027 The *file* utility shall perform a series of tests on each specified *file* in an attempt to classify it:

- 17028 1. If the file is not a regular file, its file type shall be identified. The file types directory, FIFO,
17029 socket, block special, and character special shall be identified as such. Other
17030 implementation-defined file types may also be identified.
- 17031 2. If the file is a regular file, and:
 - 17032 a. The file is zero-length, it shall be identified as an empty file.
 - 17033 b. The file is not zero-length, *file* shall examine an initial segment of the file and shall
17034 make a guess at identifying its contents or whether it is an executable binary file.
17035 (The answer is not guaranteed to be correct.)

17036 If *file* does not exist, cannot be read, or its file status could not be determined, the output shall
17037 indicate that the file was processed, but that its type could not be determined.

17038 If *file* is a symbolic link, by default the link shall be resolved and *file* shall test the type of file
17039 referenced by the symbolic link.

17040 OPTIONS

17041 The *file* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
17042 Utility Syntax Guidelines.

17043 The following options shall be supported by the implementation:

- 17044 **-d** Apply any default system tests to the file.
- 17045 **-h** When a symbolic link is encountered, identify the file as a symbolic link. If **-h** is
17046 not specified and *file* is a symbolic link that refers to a nonexistent file, *file* shall
17047 identify the file as a symbolic link, as if **-h** had been specified.
- 17048 **-i** If a file is a regular file, do not attempt to classify the type of the file further, but
17049 identify the file as specified in the STDOUT section, using a <type> string that
17050 contains the string "regular file".
- 17051 **-M file** Specify the name of a file containing tests that shall be applied to a file in order to
17052 classify it (see the EXTENDED DESCRIPTION). No default system tests shall be
17053 applied.
- 17054 **-m file** Specify the name of a file containing tests that shall be applied to a file in order to
17055 classify it (see the EXTENDED DESCRIPTION).

17056 If multiple instances of the **-m**, **-d**, or **-M** options are specified, the concatenation of the tests
17057 specified, in the order specified, shall be the set of tests that are applied. If a **-M** option is
17058 specified, no tests other than those specified using the **-d**, **-M**, and **-m** options shall be applied
17059 to the file. If neither the **-d** nor **-M** options are specified, any default system tests shall be
17060 applied after any tests specified using the **-m** option.

17061 **OPERANDS**

17062 The following operand shall be supported:

17063 *file* A pathname of a file to be tested.

17064 **STDIN**

17065 Not used.

17066 **INPUT FILES**

17067 The *file* can be any file type.

17068 **ENVIRONMENT VARIABLES**

17069 The following environment variables shall affect the execution of *file*:

17070 *LANG* Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

17074 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

17076 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

17079 *LC_MESSAGES*

17080 Determine the locale that should be used to affect the format and contents of
17081 diagnostic messages written to standard error and informative messages written to
17082 standard output.

17083 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

17084 **ASYNCHRONOUS EVENTS**

17085 Default.

17086 **STDOUT**

17087 In the POSIX locale, the following format shall be used to identify each operand, *file* specified:

17088 "%s: %s\n", <*file*>, <*type*>

17089 The values for <*type*> are unspecified, except that in the POSIX locale, if *file* is identified as one
17090 of the types listed in the following table, <*type*> shall contain (but is not limited to) the
17091 corresponding string. Each space shown in the strings shall be exactly one <space>.

17092

Table 4-8 File Utility Output Strings

	If <i>file</i> is a:	<type> shall contain the string:
17094	Directory	directory
17095	FIFO	fifo
17096	Socket	socket
17097	Block special	block special
17098	Character special	character special
17099	Executable binary	executable
17100	Empty regular file	empty
17101	Symbolic link	symbolic link to
17102	<i>ar</i> archive library (see <i>ar</i>)	archive
17103	Extended <i>cpio</i> format (see <i>pax</i>)	<i>cpio</i> archive
17104	Extended <i>tar</i> format (see ustar in <i>pax</i>)	<i>tar</i> archive
17105	Shell script	commands text
17106	C-language source	c program text
17107	FORTRAN source	fortran program text

17108 If *file* is identified as a symbolic link (see **-h**), the following alternative output format shall be used:

17110 "%s: %s %s\n", <file>, <type>, <contents of link>"

17111 If the file named by the *file* operand does not exist or cannot be read, the string "cannot open" shall be included as part of the <type> field, but this shall not be considered an error that affects the exit status. If the type of the file named by the *file* operand cannot be determined, the string "data" shall be included as part of the <type> field, but this shall not be considered an error that affects the exit status.

17116 STDERR

17117 The standard error shall be used only for diagnostic messages.

17118 OUTPUT FILES

17119 None.

17120 EXTENDED DESCRIPTION

17121 A file specified as an option-argument to the **-m** or **-M** options shall contain one test per line, which shall be applied to the file. If the test succeeds, the message field of the line shall be printed and no further tests shall be applied, with the exception that tests on immediately following lines beginning with a single '>' character shall be applied.

17125 Each line shall be composed of the following four <blank>-separated fields:

17126 *offset* An unsigned number (optionally preceded by a single '>' character) specifying the *offset*, in bytes, of the value in the file that is to be compared against the *value* field of the line. If the file is shorter than the specified offset, the test shall fail.

17129 If the *offset* begins with the character '>', the test contained in the line shall not be applied to the file unless the test on the last line for which the *offset* did not begin with a '>' was successful. By default, the *offset* shall be interpreted as an unsigned decimal number. With a leading 0x or 0X, the *offset* shall be interpreted as a hexadecimal number; otherwise, with a leading 0, the *offset* shall be interpreted as an octal number.

17135 *type* The type of the value in the file to be tested. 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.

17138 The *type* string shall be interpreted as the bytes from the file starting at the
17139 specified *offset* and including the same number of bytes specified by the *value* field.
17140 If insufficient bytes remain in the file past the *offset* to match the *value* field, the test
17141 shall fail.

17142 The type specification characters **d**, **f**, and **u** can be followed by an optional
17143 unsigned decimal integer that specifies the number of bytes represented by the
17144 type. The type specification character **f** can be followed by an optional **F**, **D**, or **L**,
17145 indicating that the value is of type **float**, **double**, or **long double**, respectively. The
17146 type specification characters **d** and **u** can be followed by an optional **C**, **S**, **I**, or **L**,
17147 indicating that the value is of type **char**, **short**, **int**, or **long**, respectively.

17148 The default number of bytes represented by the type specifiers **d**, **f**, and **u** shall
17149 correspond to their respective C-language types as follows. If the system claims
17150 conformance to the C-Language Development Utilities option, those specifiers
17151 shall correspond to the default sizes used in the *c99* utility. Otherwise, the default
17152 sizes shall be implementation-defined.

17153 For the type specifier characters **d** and **u**, the default number of bytes shall
17154 correspond to the size of a basic integer type of the implementation. For these
17155 specifier characters, the implementation shall support values of the optional
17156 number of bytes to be converted corresponding to the number of bytes in the C-
17157 language types **char**, **short**, **int**, or **long**. These numbers can also be specified by an
17158 application as the characters **C**, **S**, **I**, and **L**, respectively. The byte order used when
17159 interpreting numeric values is implementation-defined, but shall correspond to the
17160 order in which a constant of the corresponding type is stored in memory on the
17161 system.

17162 For the type specifier **f**, the default number of bytes shall correspond to the
17163 number of bytes in the basic double precision floating-point data type of the
17164 underlying implementation. The implementation shall support values of the
17165 optional number of bytes to be converted corresponding to the number of bytes in
17166 the C-language types **float**, **double**, and **long double**. These numbers can also be
17167 specified by an application as the characters **F**, **D**, and **L**, respectively.

17168 All type specifiers, except for **s**, can be followed by a mask specifier of the form
17169 **&number**. The mask value shall be AND'ed with the value of the input file before
17170 the comparison with the *value* field of the line is made. By default, the mask shall
17171 be interpreted as an unsigned decimal number. With a leading **0x** or **0X**, the mask
17172 shall be interpreted as an unsigned hexadecimal number; otherwise, with a leading
17173 **0**, the mask shall be interpreted as an unsigned octal number.

17174 The strings **byte**, **short**, **long**, and **string** shall also be supported as type fields,
17175 being interpreted as **dC**, **dS**, **dL**, and **s**, respectively.

17176 **value** The *value* to be compared with the value from the file.

17177 If the specifier from the type field is **s** or **string**, then interpret the value as a string.
17178 Otherwise, interpret it as a number. If the value is a string, then the test shall
17179 succeed only when a string value exactly matches the bytes from the file.

17180 If the *value* is a string, it can contain the following sequences:

17181 **\character** The backslash-escape sequences as specified in the Base
17182 Definitions volume of IEEE Std 1003.1-2001, Table 5-1, Escape
17183 Sequences and Associated Actions ('\\', '\a', '\b', '\f',
17184 '\n', '\r', '\t', '\v'). The results of using any other

17185		character, other than an octal digit, following the backslash are
17186		unspecified.
17187	\octal	Octal sequences that can be used to represent characters with specific coded values. An octal sequence shall consist of a backslash followed by the longest sequence of one, two, or three octal-digit characters (01234567). If the size of a byte on the system is greater than 9 bits, the valid escape sequence used to represent a byte is implementation-defined.
17188		
17189		
17190		
17191		
17192		
17193		By default, any value that is not a string shall be interpreted as a signed decimal number. Any such value, with a leading 0x or 0X, shall be interpreted as an unsigned hexadecimal number; otherwise, with a leading zero, the value shall be interpreted as an unsigned octal number.
17194		
17195		
17196		
17197		If the value is not a string, it can be preceded by a character indicating the comparison to be performed. Permissible characters and the comparisons they specify are as follows:
17198		
17199		
17200	=	The test shall succeed if the value from the file equals the <i>value</i> field.
17201	<	The test shall succeed if the value from the file is less than the <i>value</i> field.
17202	>	The test shall succeed if the value from the file is greater than the <i>value</i> field.
17203	&	The test shall succeed if all of the set bits in the <i>value</i> field are set in the value from the file.
17204		
17205	^	The test shall succeed if at least one of the set bits in the <i>value</i> field is not set in the value from the file.
17206		
17207	x	The test shall succeed if the file is large enough to contain a value of the type specified starting at the offset specified.
17208		
17209	message	The <i>message</i> to be printed if the test succeeds. The <i>message</i> shall be interpreted using the notation for the <i>printf</i> formatting specification; see <i>printf</i> . If the <i>value</i> field was a string, then the value from the file shall be the argument for the <i>printf</i> formatting specification; otherwise, the value from the file shall be the argument.
17210		
17211		
17212		

17213 EXIT STATUS

17214 The following exit values shall be returned:

17215 0 Successful completion.

17216 >0 An error occurred.

17217 CONSEQUENCES OF ERRORS

17218 Default.

17219 APPLICATION USAGE

17220 The *file* utility can only be required to guess at many of the file types because only exhaustive testing can determine some types with certainty. For example, binary data on some implementations might match the initial segment of an executable or a *tar* archive.

17221 Note that the table indicates that the output contains the stated string. Systems may add text before or after the string. For executables, as an example, the machine architecture and various facts about how the file was link-edited may be included.

17226 EXAMPLES

17227 Determine whether an argument is a binary executable file:

```
17228     file "$1" | grep -Fq executable &&
17229         printf "%s is executable.\n" "$1"
```

17230 RATIONALE

17231 The **-f** option was omitted because the same effect can (and should) be obtained using the **xargs** utility.

17233 Historical versions of the *file* utility attempt to identify the following types of files: symbolic link,
17234 directory, character special, block special, socket, *tar* archive, *cpio* archive, SCCS archive, archive
17235 library, empty, *compress* output, *pack* output, binary data, C source, FORTRAN source, assembler
17236 source, *nroff/troff/eqn/tbl* source *troff* output, shell script, C shell script, English text, ASCII text,
17237 various executables, APL workspace, compiled terminfo entries, and CURSES screen images.
17238 Only those types that are reasonably well specified in POSIX or are directly related to POSIX
17239 utilities are listed in the table.

17240 Historical systems have used a “magic file” named */etc/magic* to help identify file types. Because
17241 it is generally useful for users and scripts to be able to identify special file types, the **-m** flag and
17242 a portable format for user-created magic files has been specified. No requirement is made that an
17243 implementation of *file* use this method of identifying files, only that users be permitted to add
17244 their own classifying tests.

17245 In addition, three options have been added to historical practice. The **-d** flag has been added to
17246 permit users to cause their tests to follow any default system tests. The **-i** flag has been added to
17247 permit users to test portably for regular files in shell scripts. The **-M** flag has been added to
17248 permit users to ignore any default system tests.

17249 The historical **-c** option was omitted as not particularly useful to users or portable shell scripts.
17250 In addition, a reasonable implementation of the *file* utility would report any errors found each
17251 time the magic file is read.

17252 The historical format of the magic file was the same as that specified by the Rationale in the
17253 ISO POSIX-2:1993 standard for the *offset*, *value*, and *message* fields; however, it used less precise
17254 type fields than the format specified by the current normative text. The new type field values are
17255 a superset of the historical ones.

17256 The following is an example magic file:

```
17257 0 short    070707          cpio archive
17258 0 short    0143561         Byte-swapped cpio archive
17259 0 string   070707          ASCII cpio archive
17260 0 long     0177555         Very old archive
17261 0 short    0177545         Old archive
17262 0 short    017437          Old packed data
17263 0 string   \037\036        Packed data
17264 0 string   \377\037        Compacted data
17265 0 string   \037\235        Compressed data
17266 >2 byte&0x80 >0          Block compressed
17267 >2 byte&0x1f x           %d bits
17268 0 string   \032\001        Compiled Terminfo Entry
17269 0 short    0433           Curses screen image
17270 0 short    0434           Curses screen image
17271 0 string   <ar>          System V Release 1 archive
17272 0 string   !<arch>\n__.SYMDEF Archive random library
17273 0 string   !<arch>        Archive
```

17274 0 string ARF_BEGARF PHIGS clear text archive
17275 0 long 0x137A2950 Scalable OpenFont binary
17276 0 long 0x137A2951 Encrypted scalable OpenFont binary

17277 The use of a basic integer data type is intended to allow the implementation to choose a word size commonly used by applications on that architecture.

17279 FUTURE DIRECTIONS

17280 None.

17281 SEE ALSO

17282 *ar, ls, pax*

17283 CHANGE HISTORY

17284 First released in Issue 4.

17285 Issue 6

17286 This utility is marked as part of the User Portability Utilities option.

17287 Options and an EXTENDED DESCRIPTION are added as specified in the IEEE P1003.2b draft standard.
17288

17289 IEEE PASC Interpretations 1003.2 #192 and #178 are applied.

17290 NAME

17291 find — find files

17292 SYNOPSIS

17293 `find [-H | -L] path ... [operand_expression ...]`

17294 DESCRIPTION

17295 The *find* utility shall recursively descend the directory hierarchy from each file specified by *path*,
17296 evaluating a Boolean expression composed of the primaries described in the OPERANDS section
17297 for each file encountered.

17298 The *find* utility shall be able to descend to arbitrary depths in a file hierarchy and shall not fail
17299 due to path length limitations (unless a *path* operand specified by the application exceeds
17300 `{PATH_MAX}` requirements).

17301 The *find* utility shall detect infinite loops; that is, entering a previously visited directory that is an
17302 ancestor of the last file encountered. When it detects an infinite loop, *find* shall write a
17303 diagnostic message to standard error and shall either recover its position in the hierarchy or
17304 terminate.

17305 OPTIONS

17306 The *find* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
17307 12.2, Utility Syntax Guidelines.

17308 The following options shall be supported by the implementation:

17309 **-H** Cause the file information and file type evaluated for each symbolic link
17310 encountered on the command line to be those of the file referenced by the link, and
17311 not the link itself. If the referenced file does not exist, the file information and type
17312 shall be for the link itself. File information for all symbolic links not on the
17313 command line shall be that of the link itself.

17314 **-L** Cause the file information and file type evaluated for each symbolic link to be
17315 those of the file referenced by the link, and not the link itself.

17316 Specifying more than one of the mutually-exclusive options **-H** and **-L** shall not be considered
17317 an error. The last option specified shall determine the behavior of the utility.

17318 OPERANDS

17319 The following operands shall be supported:

17320 The *path* operand is a pathname of a starting point in the directory hierarchy.

17321 The first argument that starts with a '`-`', or is a '`!`' or a '`(`', and all subsequent arguments
17322 shall be interpreted as an *expression* made up of the following primaries and operators. In the
17323 descriptions, wherever *n* is used as a primary argument, it shall be interpreted as a decimal
17324 integer optionally preceded by a plus ('`+`') or minus ('`-`') sign, as follows:

17325 `+n` More than *n*.

17326 `n` Exactly *n*.

17327 `-n` Less than *n*.

17328 The following primaries shall be supported:

17329 **-name pattern**

17330 The primary shall evaluate as true if the basename of the filename being examined
17331 matches *pattern* using the pattern matching notation described in Section 2.13 (on
17332 page 62).

17333	-nouser	The primary shall evaluate as true if the file belongs to a user ID for which the <i>getpwuid()</i> function defined in the System Interfaces volume of IEEE Std 1003.1-2001 (or equivalent) returns NULL.
17334		
17335		
17336	-nogroup	The primary shall evaluate as true if the file belongs to a group ID for which the <i>getgrgid()</i> function defined in the System Interfaces volume of IEEE Std 1003.1-2001 (or equivalent) returns NULL.
17337		
17338		
17339	-xdev	The primary shall always evaluate as true; it shall cause <i>find</i> not to continue descending past directories that have a different device ID (<i>st_dev</i> , see the <i>stat()</i> function defined in the System Interfaces volume of IEEE Std 1003.1-2001). If any -xdev primary is specified, it shall apply to the entire expression even if the -xdev primary would not normally be evaluated.
17340		
17341		
17342		
17343		
17344	-prune	The primary shall always evaluate as true; it shall cause <i>find</i> not to descend the current pathname if it is a directory. If the -depth primary is specified, the -prune primary shall have no effect.
17345		
17346		
17347	-perm [-]mode	<p>The <i>mode</i> argument is used to represent file mode bits. It shall be identical in format to the <i>symbolic_mode</i> operand described in <i>chmod</i>, and shall be interpreted as follows. To start, a template shall be assumed with all file mode bits cleared. An <i>op</i> symbol of '+' shall set the appropriate mode bits in the template; '-' shall clear the appropriate bits; '=' shall set the appropriate mode bits, without regard to the contents of process' file mode creation mask. The <i>op</i> symbol of '-' cannot be the first character of <i>mode</i>; this avoids ambiguity with the optional leading hyphen. Since the initial mode is all bits off, there are not any symbolic modes that need to use '-' as the first character.</p> <p>If the hyphen is omitted, the primary shall evaluate as true when the file permission bits exactly match the value of the resulting template.</p> <p>Otherwise, if <i>mode</i> is prefixed by a hyphen, the primary shall evaluate as true if at least all the bits in the resulting template are set in the file permission bits.</p>
17348		
17349		
17350		
17351		
17352		
17353		
17354		
17355		
17356		
17357		
17358		
17359		
17360		
17361	-perm [-]onum	<p>If the hyphen is omitted, the primary shall evaluate as true when the file permission bits exactly match the value of the octal number <i>onum</i> and only the bits corresponding to the octal mask 07777 shall be compared. (See the description of the octal <i>mode</i> in <i>chmod</i>.) Otherwise, if <i>onum</i> is prefixed by a hyphen, the primary shall evaluate as true if at least all of the bits specified in <i>onum</i> that are also set in the octal mask 07777 are set.</p>
17362		
17363		
17364		
17365		
17366		
17367		
17368	-type c	The primary shall evaluate as true if the type of the file is <i>c</i> , where <i>c</i> is 'b', 'c', 'd', 'l', 'p', 'f', or 's' for block special file, character special file, directory, symbolic link, FIFO, regular file, or socket, respectively.
17369		
17370		
17371	-links n	The primary shall evaluate as true if the file has <i>n</i> links.
17372	-user uname	The primary shall evaluate as true if the file belongs to the user <i>uname</i> . If <i>uname</i> is a decimal integer and the <i>getpwnam()</i> (or equivalent) function does not return a valid user name, <i>uname</i> shall be interpreted as a user ID.
17373		
17374		
17375	-group gname	The primary shall evaluate as true if the file belongs to the group <i>gname</i> . If <i>gname</i> is a decimal integer and the <i>getgrnam()</i> (or equivalent) function does not return a valid group name, <i>gname</i> shall be interpreted as a group ID.
17376		
17377		
17378		

17379	-size <i>n[c]</i>	The primary shall evaluate as true if the file size in bytes, divided by 512 and rounded up to the next integer, is <i>n</i> . If <i>n</i> is followed by the character 'c', the size shall be in bytes.
17382	-atime <i>n</i>	The primary shall evaluate as true if the file access time subtracted from the initialization time, divided by 86 400 (with any remainder discarded), is <i>n</i> .
17384	-ctime <i>n</i>	The primary shall evaluate as true if the time of last change of file status information subtracted from the initialization time, divided by 86 400 (with any remainder discarded), is <i>n</i> .
17387	-mtime <i>n</i>	The primary shall evaluate as true if the file modification time subtracted from the initialization time, divided by 86 400 (with any remainder discarded), is <i>n</i> .
17389	-exec <i>utility_name</i> [<i>argument</i> ...];	
17390	-exec <i>utility_name</i> [<i>argument</i> ...] <i>{}</i> +	The end of the primary expression shall be punctuated by a semicolon or by a plus sign. Only a plus sign that follows an argument containing the two characters " <i>{}</i> " shall punctuate the end of the primary expression. Other uses of the plus sign shall not be treated as special.
17395		If the primary expression is punctuated by a semicolon, the utility <i>utility_name</i> shall be invoked once for each pathname and the primary shall evaluate as true if the utility returns a zero value as exit status. A <i>utility_name</i> or <i>argument</i> containing only the two characters " <i>{}</i> " shall be replaced by the current pathname.
17400		If the primary expression is punctuated by a plus sign, the primary shall always evaluate as true, and the pathnames for which the primary is evaluated shall be aggregated into sets. The utility <i>utility_name</i> shall be invoked once for each set of aggregated pathnames. Each invocation shall begin after the last pathname in the set is aggregated, and shall be completed before the <i>find</i> utility exits and before the first pathname in the next set (if any) is aggregated for this primary, but it is otherwise unspecified whether the invocation occurs before, during, or after the evaluations of other primaries. If any invocation returns a non-zero value as exit status, the <i>find</i> utility shall return a non-zero exit status. An argument containing only the two characters " <i>{}</i> " shall be replaced by the set of aggregated pathnames, with each pathname passed as a separate argument to the invoked utility in the same order that it was aggregated. The size of any set of two or more pathnames shall be limited such that execution of the utility does not cause the system's {ARG_MAX} limit to be exceeded. If more than one argument containing only the two characters " <i>{}</i> " is present, the behavior is unspecified.
17414		If a <i>utility_name</i> or <i>argument</i> string contains the two characters " <i>{}</i> ", but not just the two characters " <i>{}</i> ", it is implementation-defined whether <i>find</i> replaces those two characters or uses the string without change. The current directory for the invocation of <i>utility_name</i> shall be the same as the current directory when the <i>find</i> utility was started. If the <i>utility_name</i> names any of the special built-in utilities (see Section 2.14 (on page 64)), the results are undefined.
17420	-ok <i>utility_name</i> [<i>argument</i> ...];	The -ok primary shall be equivalent to -exec , except that the use of a plus sign to punctuate the end of the primary expression need not be supported, and <i>find</i> shall request affirmation of the invocation of <i>utility_name</i> using the current file as an argument by writing to standard error as described in the STDERR section. If the response on standard input is affirmative, the utility shall be invoked. Otherwise, the command shall not be invoked and the value of the -ok operand shall be false.

17427	-print	The primary shall always evaluate as true; it shall cause the current pathname to be written to standard output.
17428		
17429	-newer file	The primary shall evaluate as true if the modification time of the current file is more recent than the modification time of the file named by the pathname <i>file</i> .
17430		
17431	-depth	The primary shall always evaluate as true; it shall cause descent of the directory hierarchy to be done so that all entries in a directory are acted on before the directory itself. If a -depth primary is not specified, all entries in a directory shall be acted on after the directory itself. If any -depth primary is specified, it shall apply to the entire expression even if the -depth primary would not normally be evaluated.
17432		
17433		
17434		
17435		
17436		
17437		The primaries can be combined using the following operators (in order of decreasing precedence):
17438		
17439	(<i>expression</i>)	True if <i>expression</i> is true.
17440	! <i>expression</i>	Negation of a primary; the unary NOT operator.
17441	<i>expression</i> [-a] <i>expression</i>	
17442		Conjunction of primaries; the AND operator is implied by the juxtaposition of two primaries or made explicit by the optional -a operator. The second expression shall not be evaluated if the first expression is false.
17443		
17444		
17445	<i>expression</i> -o <i>expression</i>	
17446		Alternation of primaries; the OR operator. The second expression shall not be evaluated if the first expression is true.
17447		
17448		If no <i>expression</i> is present, -print shall be used as the expression. Otherwise, if the given expression does not contain any of the primaries -exec , -ok , or -print , the given expression shall be effectively replaced by:
17449		
17450		
17451	(<i>given_expression</i>) -print	
17452		The -user , -group , and -newer primaries each shall evaluate their respective arguments only once.
17453		
17454	STDIN	
17455		If the -ok primary is used, the response shall be read from the standard input. An entire line shall be read as the response. Otherwise, the standard input shall not be used.
17456		
17457	INPUT FILES	
17458		None.
17459	ENVIRONMENT VARIABLES	
17460		The following environment variables shall affect the execution of <i>find</i> :
17461	LANG	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
17462		
17463		
17464		
17465	LC_ALL	If set to a non-empty string value, override the values of all the other internationalization variables.
17466		
17467	LC_COLLATE	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the pattern matching notation for the -n option and in the extended regular expression defined for the yesexpr locale
17468		
17469		
17470		

17471		keyword in the <i>LC_MESSAGES</i> category.
17472	<i>LC_CTYPE</i>	This variable determines the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments), the behavior of character classes within the pattern matching notation used for the -n option, and the behavior of character classes within regular expressions used in the extended regular expression defined for the yesexpr locale keyword in the <i>LC_MESSAGES</i> category.
17478	<i>LC_MESSAGES</i>	
17479		Determine the locale for the processing of affirmative responses that should be used to affect the format and contents of diagnostic messages written to standard error.
17481		
17482 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
17483	<i>PATH</i>	Determine the location of the <i>utility_name</i> for the -exec and -ok primaries, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
17485		
17486	ASYNCHRONOUS EVENTS	
17487		Default.
17488	STDOUT	
17489		The -print primary shall cause the current pathnames to be written to standard output. The format shall be:
17490		
17491		"%s\n" , <path>
17492	STDERR	
17493		The -ok primary shall write a prompt to standard error containing at least the <i>utility_name</i> to be invoked and the current pathname. In the POSIX locale, the last non-<blank> in the prompt shall be '?'. The exact format used is unspecified.
17494		
17495		
17496		Otherwise, the standard error shall be used only for diagnostic messages.
17497	OUTPUT FILES	
17498		None.
17499	EXTENDED DESCRIPTION	
17500		None.
17501	EXIT STATUS	
17502		The following exit values shall be returned:
17503		0 All <i>path</i> operands were traversed successfully.
17504		>0 An error occurred.
17505	CONSEQUENCES OF ERRORS	
17506		Default.

17507 APPLICATION USAGE

17508 When used in operands, pattern matching notation, semicolons, opening parentheses, and
17509 closing parentheses are special to the shell and must be quoted (see Section 2.2 (on page 30)).

17510 The bit that is traditionally used for sticky (historically 01000) is specified in the **-perm** primary
17511 using the octal number argument form. Since this bit is not defined by this volume of
17512 IEEE Std 1003.1-2001, applications must not assume that it actually refers to the traditional
17513 sticky bit.

17514 EXAMPLES

17515 1. The following commands are equivalent:

17516 find .
17517 find . -print

17518 They both write out the entire directory hierarchy from the current directory.

17519 2. The following command:

17520 find / \(\ -name tmp -o -name '*.\xx' \) -atime +7 -exec rm {} \;
17521 removes all files named **tmp** or ending in **.xx** that have not been accessed for seven or more
17522 24-hour periods.

17523 3. The following command:

17524 find . -perm -o+w,+s
17525 prints (**-print** is assumed) the names of all files in or below the current directory, with all
17526 of the file permission bits S_ISUID, S_ISGID, and S_IWOTH set.

17527 4. The following command:

17528 find . -name SCCS -prune -o -print

17529 recursively prints pathnames of all files in the current directory and below, but skips
17530 directories named **SCCS** and files in them.

17531 5. The following command:

17532 find . -print -name SCCS -prune

17533 behaves as in the previous example, but prints the names of the **SCCS** directories.

17534 6. The following command is roughly equivalent to the **-nt** extension to *test*:

17535 if [-n "\\$(find file1 -prune -newer file2)"]; then
17536 printf %s\n "file1 is newer than file2"
17537 fi

17538 7. The descriptions of **-atime**, **-ctime**, and **-mtime** use the terminology *n* “86 400 second
17539 periods (days)”. For example, a file accessed at 23:59 is selected by:

17540 find . -atime -1 -print

17541 at 00:01 the next day (less than 24 hours later, not more than one day ago); the midnight
17542 boundary between days has no effect on the 24-hour calculation.

17543 RATIONALE

17544 The **-a** operator was retained as an optional operator for compatibility with historical shell
17545 scripts, even though it is redundant with expression concatenation.

17546 The descriptions of the ‘–’ modifier on the *mode* and *onum* arguments to the **–perm** primary
17547 agree with historical practice on BSD and System V implementations. System V and BSD
17548 documentation both describe it in terms of checking additional bits; in fact, it uses the same bits,
17549 but checks for having at least all of the matching bits set instead of having exactly the matching
17550 bits set.

17551 The exact format of the interactive prompts is unspecified. Only the general nature of the
17552 contents of prompts are specified because:

- Implementations may desire more descriptive prompts than those used on historical
implementations.
- Since the historical prompt strings do not terminate with <newline>s, there is no portable
way for another program to interact with the prompts of this utility via pipes.

17555 Therefore, an application using this prompting option relies on the system to provide the most
17556 suitable dialog directly with the user, based on the general guidelines specified.

17559 The **–name** *file* operand was changed to use the shell pattern matching notation so that *find* is
17560 consistent with other utilities using pattern matching.

17561 The **–size** operand refers to the size of a file, rather than the number of blocks it may occupy in
17562 the file system. The intent is that the *st_size* field defined in the System Interfaces volume of
17563 IEEE Std 1003.1-2001 should be used, not the *st_blocks* found in historical implementations. There
17564 are at least two reasons for this:

1. In both System V and BSD, *find* only uses *st_size* in size calculations for the operands
specified by this volume of IEEE Std 1003.1-2001. (BSD uses *st_blocks* only when processing
the **–ls** primary.)
2. Users usually think of file size in terms of bytes, which is also the unit used by the *ls* utility
for the output from the **–l** option. (In both System V and BSD, *ls* uses *st_size* for the **–l**
option size field and uses *st_blocks* for the *ls* **–s** calculations. This volume of
IEEE Std 1003.1-2001 does not specify *ls* **–s**.)

17572 The descriptions of **–atime**, **–ctime**, and **–mtime** were changed from the SVID description of *n*
17573 “days” to “24-hour periods”. The description is also different in terms of the exact timeframe for
17574 the *n* case (*versus* the *+n* or *-n*), but it matches all known historical implementations. It refers to
17575 one 86 400 second period in the past, not any time from the beginning of that period to the
17576 current time. For example, **–atime** 3 is true if the file was accessed any time in the period from 72
17577 hours to 48 hours ago.

17578 Historical implementations do not modify “{}” when it appears as a substring of an **–exec** or
17579 **–ok** *utility_name* or argument string. There have been numerous user requests for this extension,
17580 so this volume of IEEE Std 1003.1-2001 allows the desired behavior. At least one recent
17581 implementation does support this feature, but encountered several problems in managing
17582 memory allocation and dealing with multiple occurrences of “{}” in a string while it was being
17583 developed, so it is not yet required behavior.

17584 Assuming the presence of **–print** was added to correct a historical pitfall that plagues novice
17585 users, it is entirely upwards-compatible from the historical System V *find* utility. In its simplest
17586 form (*find directory*), it could be confused with the historical BSD fast *find*. The BSD developers
17587 agreed that adding **–print** as a default expression was the correct decision and have added the
17588 fast *find* functionality within a new utility called *locate*.

17589 Historically, the **–L** option was implemented using the primary **–follow**. The **–H** and **–L** options
17590 were added for two reasons. First, they offer a finer granularity of control and consistency with
17591 other programs that walk file hierarchies. Second, the **–follow** primary always evaluated to true.

As they were historically really global variables that took effect before the traversal began, some valid expressions had unexpected results. An example is the expression **-print -o -follow**. Because **-print** always evaluates to true, the standard order of evaluation implies that **-follow** would never be evaluated. This was never the case. Historical practice for the **-follow** primary, however, is not consistent. Some implementations always follow symbolic links on the command line whether **-follow** is specified or not. Others follow symbolic links on the command line only if **-follow** is specified. Both behaviors are provided by the **-H** and **-L** options, but scripts using the current **-follow** primary would be broken if the **-follow** option is specified to work either way.

Since the **-L** option resolves all symbolic links and the **-type l** primary is true for symbolic links that still exist after symbolic links have been resolved, the command:

```
17603 find -L . -type l
```

prints a list of symbolic links reachable from the current directory that do not resolve to accessible files.

A feature of SVR4's *find* utility was the **-exec** primary's + terminator. This allowed filenames containing special characters (especially <newline>s) to be grouped together without the problems that occur if such filenames are piped to *xargs*. Other implementations have added other ways to get around this problem, notably a **-print0** primary that wrote filenames with a null byte terminator. This was considered here, but not adopted. Using a null terminator meant that any utility that was going to process *find*'s **-print0** output had to add a new option to parse the null terminators it would now be reading.

The "**-exec ... {} +**" syntax adopted was a result of IEEE PASC Interpretation 1003.2 #210. It should be noted that this is an incompatible change to the ISO/IEC 9899:1999 standard. For example, the following command prints all files with a '-' after their name if they are regular files, and a '+' otherwise:

```
17617 find / -type f -exec echo {} - ';' -o -exec echo {} + ';'
```

The change invalidates usage like this. Even though the previous standard stated that this usage would work, in practice many did not support it and the standard developers felt it better to now state that this was not allowable.

17621 FUTURE DIRECTIONS

17622 None.

17623 SEE ALSO

17624 Section 2.2 (on page 30), Section 2.13 (on page 62), Section 2.14 (on page 64), *chmod*, *pax*, *sh*, *test*, the System Interfaces volume of IEEE Std 1003.1-2001, *getgrgid()*, *getpwuid()*, *stat()*

17626 CHANGE HISTORY

17627 First released in Issue 2.

17628 Issue 5

17629 The FUTURE DIRECTIONS section is added.

17630 Issue 6

17631 The following new requirements on POSIX implementations derive from alignment with the Single UNIX Specification:

- The **-perm [-]onum** primary is supported.

17634 The *find* utility is aligned with the IEEE P1003.2b draft standard, to include processing of symbolic links and changes to the description of the **atime**, **ctime**, and **mtime** operands.

17636

IEEE PASC Interpretation 1003.2 #210 is applied, extending the **-exec** operand.

17637 NAME

17638 fold — filter for folding lines

17639 SYNOPSIS

17640 fold [-bs][-w width][file...]

17641 DESCRIPTION

17642 The *fold* utility is a filter that shall fold lines from its input files, breaking the lines to have a maximum of *width* column positions (or bytes, if the **-b** option is specified). Lines shall be broken by the insertion of a <newline> such that each output line (referred to later in this section as a *segment*) is the maximum width possible that does not exceed the specified number of column positions (or bytes). A line shall not be broken in the middle of a character. The behavior is undefined if *width* is less than the number of columns any single character in the input would occupy.

17649 If the <carriage-return>s, <backspace>s, or <tab>s are encountered in the input, and the **-b** option is not specified, they shall be treated specially:

17651 <backspace> The current count of line width shall be decremented by one, although the count
17652 never shall become negative. The *fold* utility shall not insert a <newline>
17653 immediately before or after any <backspace>.

17654 <carriage-return>

17655 The current count of line width shall be set to zero. The *fold* utility shall not insert a
17656 <newline> immediately before or after any <carriage-return>.

17657 <tab>

17658 Each <tab> encountered shall advance the column position pointer to the next tab
 stop. Tab stops shall be at each column position *n* such that *n* modulo 8 equals 1.

17659 OPTIONS

17660 The *fold* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
17661 12.2, Utility Syntax Guidelines.

17662 The following options shall be supported:

17663 **-b** Count *width* in bytes rather than column positions.

17664 **-s** If a segment of a line contains a <blank> within the first *width* column positions (or
17665 bytes), break the line after the last such <blank> meeting the width constraints. If
17666 there is no <blank> meeting the requirements, the **-s** option shall have no effect for
17667 that output segment of the input line.

17668 **-w width** Specify the maximum line length, in column positions (or bytes if **-b** is specified).
17669 The results are unspecified if *width* is not a positive decimal number. The default
17670 value shall be 80.

17671 OPERANDS

17672 The following operand shall be supported:

17673 *file* A pathname of a text file to be folded. If no *file* operands are specified, the standard
17674 input shall be used.

17675 STDIN

17676 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES
17677 section.

17678 INPUT FILES

17679 If the **-b** option is specified, the input files shall be text files except that the lines are not limited
17680 to {LINE_MAX} bytes in length. If the **-b** option is not specified, the input files shall be text files.

17681 ENVIRONMENT VARIABLES

17682 The following environment variables shall affect the execution of *fold*:

17683 **LANG** Provide a default value for the internationalization variables that are unset or null.
17684 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
17685 Internationalization Variables for the precedence of internationalization variables
17686 used to determine the values of locale categories.)

17687 **LC_ALL** If set to a non-empty string value, override the values of all the other
17688 internationalization variables.

17689 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
17690 characters (for example, single-byte as opposed to multi-byte characters in
17691 arguments and input files), and for the determination of the width in column
17692 positions each character would occupy on a constant-width font output device.

17693 *LC_MESSAGES*

17694 Determine the locale that should be used to affect the format and contents of
17695 diagnostic messages written to standard error.

17696 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

17697 ASYNCHRONOUS EVENTS

17698 Default.

17699 STDOUT

17700 The standard output shall be a file containing a sequence of characters whose order shall be
17701 preserved from the input files, possibly with inserted <newline>s.

17702 STDERR

17703 The standard error shall be used only for diagnostic messages.

17704 OUTPUT FILES

17705 None.

17706 EXTENDED DESCRIPTION

17707 None.

17708 EXIT STATUS

17709 The following exit values shall be returned:

17710 0 All input files were processed successfully.

17711 >0 An error occurred.

17712 CONSEQUENCES OF ERRORS

17713 Default.

17714 APPLICATION USAGE

17715 The *cut* and *fold* utilities can be used to create text files out of files with arbitrary line lengths. The
17716 *cut* utility should be used when the number of lines (or records) needs to remain constant. The
17717 *fold* utility should be used when the contents of long lines need to be kept contiguous.

17718 The *fold* utility is frequently used to send text files to printers that truncate, rather than fold, lines
17719 wider than the printer is able to print (usually 80 or 132 column positions).

17720 EXAMPLES

17721 An example invocation that submits a file of possibly long lines to the printer (under the
17722 assumption that the user knows the line width of the printer to be assigned by *lp*):

17723 `fold -w 132 bigfile | lp`

17724 RATIONALE

17725 Although terminal input in canonical processing mode requires the erase character (frequently
17726 set to <backspace>) to erase the previous character (not byte or column position), terminal
17727 output is not buffered and is extremely difficult, if not impossible, to parse correctly; the
17728 interpretation depends entirely on the physical device that actually displays/prints/stores the
17729 output. In all known internationalized implementations, the utilities producing output for mixed
17730 column-width output assume that a <backspace> backs up one column position and outputs
17731 enough <backspace>s to return to the start of the character when <backspace> is used to
17732 provide local line motions to support underlining and emboldening operations. Since *fold*
17733 without the **-b** option is dealing with these same constraints, <backspace> is always treated as
17734 backing up one column position rather than backing up one character.

17735 Historical versions of the *fold* utility assumed 1 byte was one character and occupied one column
17736 position when written out. This is no longer always true. Since the most common usage of *fold* is
17737 believed to be folding long lines for output to limited-length output devices, this capability was
17738 preserved as the default case. The **-b** option was added so that applications could *fold* files with
17739 arbitrary length lines into text files that could then be processed by the standard utilities. Note
17740 that although the width for the **-b** option is in bytes, a line is never split in the middle of a
17741 character. (It is unspecified what happens if a width is specified that is too small to hold a single
17742 character found in the input followed by a <newline>.)

17743 The tab stops are hardcoded to be every eighth column to meet historical practice. No new
17744 method of specifying other tab stops was invented.

17745 FUTURE DIRECTIONS

17746 None.

17747 SEE ALSO

17748 *cut*

17749 CHANGE HISTORY

17750 First released in Issue 4.

17751 Issue 6

17752 The normative text is reworded to avoid use of the term “must” for application requirements.

17753 NAME

17754 fort77 — FORTRAN compiler (**FORTRAN**)

17755 SYNOPSIS

```
17756 FD        fort77 [-c][-g][-L directory]... [-O optlevel][-o outfile][-s][-w]
17757            operand...
17758
```

17759 DESCRIPTION

17760 The *fort77* utility is the interface to the FORTRAN compilation system; it shall accept the full
 17761 FORTRAN-77 language defined by the ANSI X3.9-1978 standard. The system conceptually
 17762 consists of a compiler and link editor. The files referenced by *operands* are compiled and linked
 17763 to produce an executable file. It is unspecified whether the linking occurs entirely within the
 17764 operation of *fort77*; some implementations may produce objects that are not fully resolved until
 17765 the file is executed.

17766 If the **-c** option is present, for all pathname operands of the form *file.f*, the files:

17767 \$(basename *pathname.f*).o

17768 shall be created or overwritten as the result of successful compilation. If the **-c** option is not
 17769 specified, it is unspecified whether such .o files are created or deleted for the *file.f* operands.

17770 If there are no options that prevent link editing (such as **-c**) and all operands compile and link
 17771 without error, the resulting executable file shall be written into the file named by the **-o** option
 17772 (if present) or to the file **a.out**. The executable file shall be created as specified in the System
 17773 Interfaces volume of IEEE Std 1003.1-2001, except that the file permissions shall be set to:

17774 S_IRWXO | S_IRWXG | S_IROWXU

17775 and that the bits specified by the *umask* of the process shall be cleared.

17776 OPTIONS

17777 The *fort77* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
 17778 12.2, Utility Syntax Guidelines, except that:

- 17779 • The **-l** *library* operands have the format of options, but their position within a list of
 17780 operands affects the order in which libraries are searched.
- 17781 • The order of specifying the multiple **-L** options is significant.
- 17782 • Conforming applications shall specify each option separately; that is, grouping option letters
 17783 (for example, **-cg**) need not be recognized by all implementations.

17784 The following options shall be supported:

- | | |
|--------------------------------|---|
| 17785 -c | Suppress the link-edit phase of the compilation, and do not remove any object files
17786 that are produced. |
| 17787 -g | Produce symbolic information in the object or executable files; the nature of this
17788 information is unspecified, and may be modified by implementation-defined
17789 interactions with other options. |
| 17790 -s | Produce object or executable files, or both, from which symbolic and other
17791 information not required for proper execution using the <i>exec</i> family of functions
17792 defined in the System Interfaces volume of IEEE Std 1003.1-2001 has been removed
17793 (stripped). If both -g and -s options are present, the action taken is unspecified. |
| 17794 -o <i>outfile</i> | Use the pathname <i>outfile</i> , instead of the default a.out , for the executable file
17795 produced. If the -o option is present with -c , the result is unspecified. |

17796 **-L directory** Change the algorithm of searching for the libraries named in **-I** operands to look in the directory named by the *directory* pathname before looking in the usual places. Directories named in **-L** options shall be searched in the specified order. At least ten instances of this option shall be supported in a single *fort77* command invocation. If a directory specified by a **-L** option contains a file named **libf.a**, the results are unspecified.

17802 **-O optlevel** Specify the level of code optimization. If the *optlevel* option-argument is the digit '0', all special code optimizations shall be disabled. If it is the digit '1', the nature of the optimization is unspecified. If the **-O** option is omitted, the nature of the system's default optimization is unspecified. It is unspecified whether code generated in the presence of the **-O 0** option is the same as that generated when **-O** is omitted. Other *optlevel* values may be supported.

17808 **-w** Suppress warnings.

17809 Multiple instances of **-L** options can be specified.

17810 OPERANDS

17811 An *operand* is either in the form of a pathname or the form **-I library**. At least one operand of the pathname form shall be specified. The following operands shall be supported:

17813 **file.f** The pathname of a FORTRAN source file to be compiled and optionally passed to the link editor. The filename operand shall be of this form if the **-c** option is used.

17815 **file.a** A library of object files typically produced by *ar*, and passed directly to the link editor. Implementations may recognize implementation-defined suffixes other than **.a** as denoting object file libraries.

17818 **file.o** An object file produced by *fort77 -c* and passed directly to the link editor. Implementations may recognize implementation-defined suffixes other than **.o** as denoting object files.

17821 The processing of other files is implementation-defined.

17822 **-I library** (The letter ell.) Search the library named:
17823 **liblibrary.a**

17824 A library is searched when its name is encountered, so the placement of a **-I** operand is significant. Several standard libraries can be specified in this manner, as described in the EXTENDED DESCRIPTION section. Implementations may recognize implementation-defined suffixes other than **.a** as denoting libraries.

17828 STDIN

17829 Not used.

17830 INPUT FILES

17831 The input file shall be one of the following: a text file containing FORTRAN source code; an object file in the format produced by *fort77 -c*; or a library of object files, in the format produced by archiving zero or more object files, using *ar*. Implementations may supply additional utilities that produce files in these formats. Additional input files are implementation-defined.

17835 A <tab> encountered within the first six characters on a line of source code shall cause the compiler to interpret the following character as if it were the seventh character on the line (that is, in column 7).

17838 ENVIRONMENT VARIABLES

17839 The following environment variables shall affect the execution of *fort77*:

17840	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
17844	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
17846	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).
17849	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
17852 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
17853	<i>TMPDIR</i>	Determine the pathname that should override the default directory for temporary files, if any.

17855 ASYNCHRONOUS EVENTS

17856 Default.

17857 STDOUT

17858 Not used.

17859 STDERR

17860 The standard error shall be used only for diagnostic messages. If more than one *file* operand
17861 ending in .f (or possibly other unspecified suffixes) is given, for each such file:

17862 "%s:\n", <file>

17863 may be written to allow identification of the diagnostic message with the appropriate input file.

17864 This utility may produce warning messages about certain conditions that do not warrant
17865 returning an error (non-zero) exit value.

17866 OUTPUT FILES

17867 Object files, listing files, and executable files shall be produced in unspecified formats.

17868 EXTENDED DESCRIPTION

17869 Standard Libraries

17870 The *fort77* utility shall recognize the following -I operand for the standard library:

17871 -I f This library contains all functions referenced in the ANSI X3.9-1978 standard. This
17872 operand shall not be required to be present to cause a search of this library.

17873 In the absence of options that inhibit invocation of the link editor, such as -c, the *fort77* utility
17874 shall cause the equivalent of a -I f operand to be passed to the link editor as the last -I operand,
17875 causing it to be searched after all other object files and libraries are loaded.

17876 It is unspecified whether the library libf.a exists as a regular file. The implementation may
17877 accept as -I operands names of objects that do not exist as regular files.

17878 External Symbols

17879 The FORTRAN compiler and link editor shall support the significance of external symbols up to
17880 a length of at least 31 bytes; case folding is permitted. The action taken upon encountering
17881 symbols exceeding the implementation-defined maximum symbol length is unspecified.

17882 The compiler and link editor shall support a minimum of 511 external symbols per source or
17883 object file, and a minimum of 4 095 external symbols total. A diagnostic message is written to
17884 standard output if the implementation-defined limit is exceeded; other actions are unspecified.

17885 EXIT STATUS

17886 The following exit values shall be returned:

17887 0 Successful compilation or link edit.

17888 >0 An error occurred.

17889 CONSEQUENCES OF ERRORS

17890 When *fort77* encounters a compilation error, it shall write a diagnostic to standard error and
17891 continue to compile other source code operands. It shall return a non-zero exit status, but it is
17892 implementation-defined whether an object module is created. If the link edit is unsuccessful, a
17893 diagnostic message shall be written to standard error, and *fort77* shall exit with a non-zero
17894 status.

17895 APPLICATION USAGE

17896 None.

17897 EXAMPLES

17898 The following usage example compiles **xyz.f** and creates the executable file **foo**:

17899 **fort77 -o foo xyz.f**

17900 The following example compiles **xyz.f** and creates the object file **xyz.o**:

17901 **fort77 -c xyz.f**

17902 The following example compiles **xyz.f** and creates the executable file **a.out**:

17903 **fort77 xyz.f**

17904 The following example compiles **xyz.f**, links it with **b.o**, and creates the executable **a.out**:

17905 **fort77 xyz.f b.o**

17906 RATIONALE

17907 The name of this utility was chosen as *fort77* to parallel the renaming of the C compiler. The
17908 name *f77* was not chosen to avoid problems with historical implementations. The
17909 ANSI X3.9-1978 standard was selected as a normative reference because the ISO/IEC version of
17910 FORTRAN-77 has been superseded by the ISO/IEC 1539:1990 standard (Fortran-90).

17911 The file inclusion and symbol definition **#define** mechanisms used by the *c99* utility were not
17912 included in this volume of IEEE Std 1003.1-2001—even though they are commonly
17913 implemented—since there is no requirement that the FORTRAN compiler use the C
17914 preprocessor.

17915 The **-onetrip** option was not included in this volume of IEEE Std 1003.1-2001, even though many
17916 historical compilers support it, because it is derived from FORTRAN-66; it is an anachronism
17917 that should not be perpetuated.

17918 Some implementations produce compilation listings. This aspect of FORTRAN has been left
17919 unspecified because there was controversy concerning the various methods proposed for
17920 implementing it: a **-V** option overlapped with historical vendor practice and a naming

17921 convention of creating files with .I suffixes collided with historical *lex* file naming practice.

17922 There is no -I option in this version of this volume of IEEE Std 1003.1-2001 to specify a directory
17923 for file inclusion. An INCLUDE directive has been a part of the Fortran-90 discussions, but an
17924 interface supporting that standard is not in the current scope.

17925 It is noted that many FORTRAN compilers produce an object module even when compilation
17926 errors occur; during a subsequent compilation, the compiler may patch the object module rather
17927 than recompiling all the code. Consequently, it is left to the implementor whether or not an
17928 object file is created.

17929 A reference to MIL-STD-1753 was removed from an early proposal in response to a request from
17930 the POSIX FORTRAN-binding standard developers. It was not the intention of the standard
17931 developers to require certification of the FORTRAN compiler, and IEEE Std 1003.9-1992 does not
17932 specify the military standard or any special preprocessing requirements. Furthermore, use of
17933 that document would have been inappropriate for an international standard.

17934 The specification of optimization has been subject to changes through early proposals. At one
17935 time, -O and -N were Booleans: optimize and do not optimize (with an unspecified default).
17936 Some historical practice led this to be changed to:

17937 -O 0 No optimization.
17938 -O 1 Some level of optimization.
17939 -O n Other, unspecified levels of optimization.

17940 It is not always clear whether “good code generation” is the same thing as optimization. Simple
17941 optimizations of local actions do not usually affect the semantics of a program. The -O 0 option
17942 has been included to accommodate the very particular nature of scientific calculations in a
17943 highly optimized environment; compilers make errors. Some degree of optimization is expected,
17944 even if it is not documented here, and the ability to shut it off completely could be important
17945 when porting an application. An implementation may treat -O 0 as “do less than normal” if it
17946 wishes, but this is only meaningful if any of the operations it performs can affect the semantics
17947 of a program. It is highly dependent on the implementation whether doing less than normal is
17948 logical. It is not the intent of the -O 0 option to ask for inefficient code generation, but rather to
17949 assure that any semantically visible optimization is suppressed.

17950 The specification of standard library access is consistent with the C compiler specification.
17951 Implementations are not required to have /usr/lib/libf.a, as many historical implementations do,
17952 but if not they are required to recognize f as a token.

17953 External symbol size limits are in normative text; conforming applications need to know these
17954 limits. However, the minimum maximum symbol length should be taken as a constraint on a
17955 conforming application, not on an implementation, and consequently the action taken for a
17956 symbol exceeding the limit is unspecified. The minimum size for the external symbol table was
17957 added for similar reasons.

17958 The CONSEQUENCES OF ERRORS section clearly specifies the behavior of the compiler when
17959 compilation or link-edit errors occur. The behavior of several historical implementations was
17960 examined, and the choice was made to be silent on the status of the executable, or a.out, file in
17961 the face of compiler or linker errors. If a linker writes the executable file, then links it on disk
17962 with lseek()s and write()s, the partially linked executable file can be left on disk and its execute
17963 bits turned off if the link edit fails. However, if the linker links the image in memory before
17964 writing the file to disk, it need not touch the executable file (if it already exists) because the link
17965 edit fails. Since both approaches are historical practice, a conforming application shall rely on
17966 the exit status of fort77, rather than on the existence or mode of the executable file.

17967 The **-g** and **-s** options are not specified as mutually-exclusive. Historically these two options
17968 have been mutually-exclusive, but because both are so loosely specified, it seemed appropriate
17969 to leave their interaction unspecified.

17970 The requirement that conforming applications specify compiler options separately is to reserve
17971 the multi-character option name space for vendor-specific compiler options, which are known to
17972 exist in many historical implementations. Implementations are not required to recognize, for
17973 example, **-gc** as if it were **-g -c**; nor are they forbidden from doing so. The SYNOPSIS shows all
17974 of the options separately to highlight this requirement on applications.

17975 Echoing filenames to standard error is considered a diagnostic message because it would
17976 otherwise be difficult to associate an error message with the erring file. They are described with
17977 “may” to allow implementations to use other methods of identifying files and to parallel the
17978 description in *c99*.

17979 **FUTURE DIRECTIONS**

17980 A compilation system based on the ISO/IEC 1539:1990 standard (Fortran-90) may be considered
17981 for a future version; it may have a different utility name from *fort77*.

17982 **SEE ALSO**

17983 *ar, asa, c99, umask*, the System Interfaces volume of IEEE Std 1003.1-2001, *exec*

17984 **CHANGE HISTORY**

17985 First released in Issue 4.

17986 **Issue 6**

17987 This utility is marked as part of the FORTRAN Development Utilities option.

17988 The normative text is reworded to avoid use of the term “must” for application requirements.

17989 NAME

17990 fuser — list process IDs of all processes that have one or more files open

17991 SYNOPSIS

17992 XSI fuser [-cfu] file ...

17993

17994 DESCRIPTION

17995 The *fuser* utility shall write to standard output the process IDs of processes running on the local
17996 system that have one or more named files open. For block special devices, all processes using
17997 any file on that device are listed.

17998 The *fuser* utility shall write to standard error additional information about the named files
17999 indicating how the file is being used.

18000 Any output for processes running on remote systems that have a named file open is unspecified.

18001 A user may need appropriate privilege to invoke the *fuser* utility.

18002 OPTIONS

18003 The *fuser* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
18004 12.2, Utility Syntax Guidelines.

18005 The following options shall be supported:

18006 **-c** The file is treated as a mount point and the utility shall report on any files open in
18007 the file system.

18008 **-f** The report shall be only for the named files.

18009 **-u** The user name, in parentheses, associated with each process ID written to standard
18010 output shall be written to standard error.

18011 OPERANDS

18012 The following operand shall be supported:

18013 *file* A pathname on which the file or file system is to be reported.

18014 STDIN

18015 Not used.

18016 INPUT FILES

18017 The user database.

18018 ENVIRONMENT VARIABLES

18019 The following environment variables shall affect the execution of *fuser*:

18020 **LANG** Provide a default value for the internationalization variables that are unset or null.
18021 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
18022 Internationalization Variables for the precedence of internationalization variables
18023 used to determine the values of locale categories.)

18024 **LC_ALL** If set to a non-empty string value, override the values of all the other
18025 internationalization variables.

18026 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
18027 characters (for example, single-byte as opposed to multi-byte characters in
18028 arguments).

18029 LC_MESSAGES

18030 Determine the locale that should be used to affect the format and contents of
18031 diagnostic messages written to standard error.

18032 **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

18033 **ASYNCHRONOUS EVENTS**

18034 Default.

18035 **STDOUT**

18036 The *fuser* utility shall write the process ID for each process using each file given as an operand to
18037 standard output in the following format:

18038 "%d", <process_id>

18039 **STDERR**

18040 The *fuser* utility shall write diagnostic messages to standard error.

18041 The *fuser* utility also shall write the following to standard error:

18042 • The pathname of each named file is written followed immediately by a colon.

18043 • For each process ID written to standard output, the character 'c' shall be written to
18044 standard error if the process is using the file as its current directory and the character 'r'
18045 shall be written to standard error if the process is using the file as its root directory.
18046 Implementations may write other alphabetic characters to indicate other uses of files.

18047 • When the **-u** option is specified, characters indicating the use of the file shall be followed
18048 immediately by the user name, in parentheses, corresponding to the process' real user ID. If
18049 the user name cannot be resolved from the process' real user ID, the process' real user ID
18050 shall be written instead of the user name.

18051 When standard output and standard error are directed to the same file, the output shall be
18052 interleaved so that the filename appears at the start of each line, followed by the process ID and
18053 characters indicating the use of the file. Then, if the **-u** option is specified, the user name or user
18054 ID for each process using that file shall be written.

18055 A <newline> shall be written to standard error after the last output described above for each *file*
18056 operand.

18057 **OUTPUT FILES**

18058 None.

18059 **EXTENDED DESCRIPTION**

18060 None.

18061 **EXIT STATUS**

18062 The following exit values shall be returned:

18063 0 Successful completion.

18064 >0 An error occurred.

18065 **CONSEQUENCES OF ERRORS**

18066 Default.

18067 APPLICATION USAGE

18068 None.

18069 EXAMPLES

18070 The command:

18071 `fuser -fu .`

18072 writes to standard output the process IDs of processes that are using the current directory and
18073 writes to standard error an indication of how those processes are using the directory and the
18074 user names associated with the processes that are using the current directory.

18075 RATIONALE

18076 The definition of the *fuser* utility follows existing practice.

18077 FUTURE DIRECTIONS

18078 None.

18079 SEE ALSO

18080 None.

18081 CHANGE HISTORY

18082 First released in Issue 5.

18083 **NAME**

18084 gencat — generate a formatted message catalog

18085 **SYNOPSIS**18086 XSI gencat *catfile msgfile...*

18087

18088 **DESCRIPTION**

18089 The *gencat* utility shall merge the message text source file *msgfile* into a formatted message
18090 catalog *catfile*. The file *catfile* shall be created if it does not already exist. If *catfile* does exist, its
18091 messages shall be included in the new *catfile*. If set and message numbers collide, the new
18092 message text defined in *msgfile* shall replace the old message text currently contained in *catfile*.

18093 **OPTIONS**

18094 None.

18095 **OPERANDS**

18096 The following operands shall be supported:

18097 *catfile* A pathname of the formatted message catalog. If ‘–’ is specified, standard output
18098 shall be used. The format of the message catalog produced is unspecified.

18099 *msgfile* A pathname of a message text source file. If ‘–’ is specified for an instance of
18100 *msgfile*, standard input shall be used. The format of message text source files is
18101 defined in the EXTENDED DESCRIPTION section.

18102 **STDIN**18103 The standard input shall not be used unless a *msgfile* operand is specified as ‘–’.18104 **INPUT FILES**

18105 The input files shall be text files.

18106 **ENVIRONMENT VARIABLES**18107 The following environment variables shall affect the execution of *gencat*:

18108 *LANG* Provide a default value for the internationalization variables that are unset or null.
18109 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
18110 Internationalization Variables for the precedence of internationalization variables
18111 used to determine the values of locale categories.)

18112 *LC_ALL* If set to a non-empty string value, override the values of all the other
18113 internationalization variables.

18114 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
18115 characters (for example, single-byte as opposed to multi-byte characters in
18116 arguments and input files).

18117 *LC_MESSAGES*

18118 Determine the locale that should be used to affect the format and contents of
18119 diagnostic messages written to standard error.

18120 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.18121 **ASYNCHRONOUS EVENTS**

18122 Default.

18123 **STDOUT**18124 The standard output shall not be used unless the *catfile* operand is specified as ‘–’.

18125 STDERR

18126 The standard error shall be used only for diagnostic messages.

18127 OUTPUT FILES

18128 None.

18129 EXTENDED DESCRIPTION

18130 The content of a message text file shall be in the format defined as follows. Note that the fields of
18131 a message text source line are separated by a single <blank>. Any other <blank>s are considered
18132 to be part of the subsequent field.

18133 \$set n comment

18134 This line specifies the set identifier of the following messages until the next \$set or
18135 end-of-file appears. The *n* denotes the set identifier, which is defined as a number
18136 in the range [1, {NL_SETMAX}] (see the <limits.h> header defined in the Base
18137 Definitions volume of IEEE Std 1003.1-2001). The application shall ensure that set
18138 identifiers are presented in ascending order within a single source file, but need
18139 not be contiguous. Any string following the set identifier shall be treated as a
18140 comment. If no \$set directive is specified in a message text source file, all messages
18141 shall be located in an implementation-defined default message set NL_SETD (see
18142 the <nl_types.h> header defined in the Base Definitions volume of
18143 IEEE Std 1003.1-2001).

18144 \$delset n comment

18145 This line deletes message set *n* from an existing message catalog. The *n* denotes the
18146 set number [1, {NL_SETMAX}]. Any string following the set number shall be
18147 treated as a comment.

18148 \$ comment A line beginning with '\$' followed by a <blank> shall be treated as a comment.**18149 m message-text**

18150 The *m* denotes the message identifier, which is defined as a number in the range [1,
18151 {NL_MSGMAX}] (see the <limits.h> header). The *message-text* shall be stored in the
18152 message catalog with the set identifier specified by the last \$set directive, and with
18153 message identifier *m*. If the *message-text* is empty, and a <blank> field separator is
18154 present, an empty string shall be stored in the message catalog. If a message source
18155 line has a message number, but neither a field separator nor *message-text*, the
18156 existing message with that number (if any) shall be deleted from the catalog. The
18157 application shall ensure that message identifiers are in ascending order within a
18158 single set, but need not be contiguous. The application shall ensure that the length
18159 of *message-text* is in the range [0, {NL_TEXTMAX}] (see the <limits.h> header).

18160 \$quote n

18161 This line specifies an optional quote character *c*, which can be used to surround
18162 *message-text* so that trailing spaces or null (empty) messages are visible in a
18163 message source line. By default, or if an empty \$quote directive is supplied, no
quoting of *message-text* shall be recognized.

18164 Empty lines in a message text source file shall be ignored. The effects of lines starting with any
18165 character other than those defined above are implementation-defined.

18166 Text strings can contain the special characters and escape sequences defined in the following
18167 table:

18168
18169
18170
18171
18172
18173
18174
18175
18176
18177

Description	Symbol	Sequence
<newline>	NL(LF)	\n
Horizontal-tab	HT	\t
<vertical-tab>	VT	\v
<backspace>	BS	\b
<carriage-return>	CR	\r
<form-feed>	FF	\f
Backslash	\	\\
Bit pattern	ddd	\ddd

18178 The escape sequence "\ddd" consists of backslash followed by one, two, or three octal digits,
18179 which shall be taken to specify the value of the desired character. If the character following a
18180 backslash is not one of those specified, the backslash shall be ignored.

18181 Backslash ('\'') followed by a <newline> is also used to continue a string on the following line.
18182 Thus, the following two lines describe a single message string:

18183 1 This line continues \
18184 to the next line

18185 which shall be equivalent to:

18186 1 This line continues to the next line

18187 EXIT STATUS

18188 The following exit values shall be returned:

18189 0 Successful completion.
18190 >0 An error occurred.

18191 CONSEQUENCES OF ERRORS

18192 Default.

18193 APPLICATION USAGE

18194 Message catalogs produced by *gencat* are binary encoded, meaning that their portability cannot
18195 be guaranteed between different types of machine. Thus, just as C programs need to be
18196 recompiled for each type of machine, so message catalogs must be recreated via *gencat*.

18197 EXAMPLES

18198 None.

18199 RATIONALE

18200 None.

18201 FUTURE DIRECTIONS

18202 None.

18203 SEE ALSO

18204 *iconv*, the Base Definitions volume of IEEE Std 1003.1-2001, <limits.h>, <nl_types.h>

18205 CHANGE HISTORY

18206 First released in Issue 3.

18207 Issue 6

18208 The normative text is reworded to avoid use of the term "must" for application requirements.

18209 NAME

18210 get — get a version of an SCCS file (**DEVELOPMENT**)

18211 SYNOPSIS

18212 XSI get [-begkmnlLpst][-c cutoff][-i list][-r SID][-x list] file...

18213

18214 DESCRIPTION

18215 The *get* utility shall generate a text file from each named SCCS *file* according to the specifications
18216 given by its options.

18217 The generated text shall normally be written into a file called the **g-file** whose name is derived
18218 from the SCCS filename by simply removing the leading "s".

18219 OPTIONS

18220 The *get* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
18221 Utility Syntax Guidelines.

18222 The following options shall be supported:

18223 **-r SID** Indicate the SCCS Identification String (SID) of the version (delta) of an SCCS file
18224 to be retrieved. The table shows, for the most useful cases, what version of an
18225 SCCS file is retrieved (as well as the SID of the version to be eventually created by
18226 *delta* if the **-e** option is also used), as a function of the SID specified.

18227 **-c cutoff** Indicate the *cutoff* date-time, in the form:

18228 YY[MM[DD[HH[MM[SS]]]]]

18229 For the YY component, values in the range [69,99] shall refer to years 1969 to 1999
18230 inclusive, and values in the range [00,68] shall refer to years 2000 to 2068 inclusive.

18231 **Note:** It is expected that in a future version of IEEE Std 1003.1-2001 the default
18232 century inferred from a 2-digit year will change. (This would apply to all
18233 commands accepting a 2-digit year as input.)

18234 No changes (deltas) to the SCCS file that were created after the specified *cutoff*
18235 date-time shall be included in the generated text file. Units omitted from the date-
18236 time default to their maximum possible values; for example, **-c 7502** is equivalent
18237 to **-c 750228235959**.

18238 Any number of non-numeric characters may separate the various 2-digit pieces of
18239 the *cutoff* date-time. This feature allows the user to specify a *cutoff* date in the form:
18240 **-c "77/2/2 9:22:25"**.

18241 **-e** Indicate that the *get* is for the purpose of editing or making a change (delta) to the
18242 SCCS file via a subsequent use of *delta*. The **-e** option used in a *get* for a particular
18243 version (SID) of the SCCS file shall prevent further *get* commands from editing on
18244 the same SID until *delta* is executed or the **j** (joint edit) flag is set in the SCCS file.
18245 Concurrent use of *get -e* for different SIDs is always allowed.

18246 If the **g-file** generated by *get* with a **-e** option is accidentally ruined in the process
18247 of editing, it may be regenerated by re-executing the *get* command with the **-k**
18248 option in place of the **-e** option.

18249 SCCS file protection specified via the ceiling, floor, and authorized user list stored
18250 in the SCCS file shall be enforced when the **-e** option is used.

18251 **-b** Use with the **-e** option to indicate that the new delta should have an SID in a new
18252 branch as shown in the table below. This option shall be ignored if the **b** flag is not
18253 present in the file or if the retrieved delta is not a leaf delta. (A leaf delta is one that

18254 has no successors on the SCCS file tree.)

18255 **Note:** A branch delta may always be created from a non-leaf delta.

18256 **-i list** Indicate a *list* of deltas to be included (forced to be applied) in the creation of the generated file. The *list* has the following syntax:

```
18258                   <list> ::= <range> | <list> , <range>
18259                   <range> ::= SID | SID - SID
```

18260 SID, the SCCS Identification of a delta, may be in any form shown in the “SID Specified” column of the table in the EXTENDED DESCRIPTION section, except that the result of supplying a partial SID is unspecified. A diagnostic message shall be written if the first SID in the range is not an ancestor of the second SID in the range.

18265 **-x list** Indicate a *list* of deltas to be excluded (forced not to be applied) in the creation of the generated file. See the **-i** option for the *list* format.

18266

18267 **-k** Suppress replacement of identification keywords (see below) in the retrieved text by their value. The **-k** option shall be implied by the **-e** option.

18268

18269 **-l** Write a delta summary into an **l-file**.

18270 **-L** Write a delta summary to standard output. All informative output that normally is written to standard output shall be written to standard error instead, unless the **-s** option is used, in which case it shall be suppressed.

18271

18272

18273 **-p** Write the text retrieved from the SCCS file to the standard output. No **g-file** shall be created. All informative output that normally goes to the standard output shall go to standard error instead, unless the **-s** option is used, in which case it shall disappear.

18274

18275

18276

18277 **-s** Suppress all informative output normally written to standard output. However, fatal error messages (which shall always be written to the standard error) shall remain unaffected.

18278

18279

18280 **-m** Precede each text line retrieved from the SCCS file by the SID of the delta that inserted the text line in the SCCS file. The format shall be:

```
18282                   "%s\t%s", <SID>, <text line>
```

18283 **-n** Precede each generated text line with the %M% identification keyword value (see below). The format shall be:

```
18285                   "%s\t%s", <%M% value>, <text line>
```

18286 When both the **-m** and **-n** options are used, the *<text line>* shall be replaced by the **-m** option-generated format.

18287

18288 **-g** Suppress the actual retrieval of text from the SCCS file. It is primarily used to generate an **l-file**, or to verify the existence of a particular SID.

18289

18290 **-t** Use to access the most recently created (top) delta in a given release (for example, **-r 1**), or release and level (for example, **-r 1.2**).

18291

18292 OPERANDS

18293 The following operands shall be supported:

18294 **file** A pathname of an existing SCCS file or a directory. If *file* is a directory, the *get* utility shall behave as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the pathname does not begin

18297 with **s**.) and unreadable files shall be silently ignored.

18298 If exactly one *file* operand appears, and it is '**-**', the standard input shall be read;

18299 each line of the standard input is taken to be the name of an SCCS file to be

18300 processed. Non-SCCS files and unreadable files shall be silently ignored.

18301 **STDIN**

18302 The standard input shall be a text file used only if the *file* operand is specified as '**-**'. Each line

18303 of the text file shall be interpreted as an SCCS pathname.

18304 **INPUT FILES**

18305 The SCCS files shall be files of an unspecified format.

18306 **ENVIRONMENT VARIABLES**

18307 The following environment variables shall affect the execution of *get*:

18308 **LANG** Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
Internationalization Variables for the precedence of internationalization variables
used to determine the values of locale categories.)

18312 **LC_ALL** If set to a non-empty string value, override the values of all the other
internationalization variables.

18314 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
characters (for example, single-byte as opposed to multi-byte characters in
arguments and input files).

18317 **LC_MESSAGES**

18318 Determine the locale that should be used to affect the format and contents of
18319 diagnostic messages written to standard error, and informative messages written
18320 to standard output (or standard error, if the **-p** option is used).

18321 **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

18322 **TZ** Determine the timezone in which the times and dates written in the SCCS file are
evaluated. If the **TZ** variable is unset or NULL, an unspecified system default
timezone is used.

18325 **ASYNCHRONOUS EVENTS**

18326 Default.

18327 **STDOUT**

18328 For each file processed, *get* shall write to standard output the SID being accessed and the number
18329 of lines retrieved from the SCCS file, in the following format:

18330 "%s\n%d lines\n", <SID>, <number of lines>

18331 If the **-e** option is used, the SID of the delta to be made shall appear after the SID accessed and
18332 before the number of lines generated, in the POSIX locale:

18333 "%s\nnew delta %s\n%d lines\n", <SID accessed>,
18334 <SID to be made>, <number of lines>

18335 If there is more than one named file or if a directory or standard input is named, each pathname
18336 shall be written before each of the lines shown in one of the preceding formats:

18337 "\n%s:\n", <pathname>

18338 If the **-L** option is used, a delta summary shall be written following the format specified below
18339 for **l-files**.

18340 If the **-i** option is used, included deltas shall be listed following the notation, in the POSIX locale:
18341 "Included:\n"
18342 If the **-x** option is used, excluded deltas shall be listed following the notation, in the POSIX
18343 locale:
18344 "Excluded:\n"
18345 If the **-p** or **-L** options are specified, the standard output shall consist of the text retrieved from
18346 the SCCS file.

18347 **STDERR**

18348 The standard error shall be used only for diagnostic messages, except if the **-p** or **-L** options are
18349 specified, it shall include all informative messages normally sent to standard output.

18350 **OUTPUT FILES**

18351 Several auxiliary files may be created by *get*. These files are known generically as the **g-file**, **l-**
18352 **file**, **p-file**, and **z-file**. The letter before the hyphen is called the *tag*. An auxiliary filename shall
18353 be formed from the SCCS filename: the application shall ensure that the last component of all
18354 SCCS filenames is of the form **s.module-name**; the auxiliary files shall be named by replacing the
18355 leading **s** with the tag. The **g-file** shall be an exception to this scheme: the **g-file** is named by
18356 removing the **s**. prefix. For example, for **s.xyz.c**, the auxiliary filenames would be **xyz.c**, **l.xyz.c**,
18357 **p.xyz.c**, and **z.xyz.c**, respectively.

18358 The **g-file**, which contains the generated text, shall be created in the current directory (unless the
18359 **-p** option is used). A **g-file** shall be created in all cases, whether or not any lines of text were
18360 generated by the *get*. It shall be owned by the real user. If the **-k** option is used or implied, the
18361 **g-file** shall be writable by the owner only (read-only for everyone else); otherwise, it shall be
18362 read-only. Only the real user need have write permission in the current directory.

18363 The **l-file** shall contain a table showing which deltas were applied in generating the retrieved
18364 text. The **l-file** shall be created in the current directory if the **-l** option is used; it shall be read-
18365 only and it is owned by the real user. Only the real user need have write permission in the
18366 current directory.

18367 Lines in the **l-file** shall have the following format:

18368 "%c%c%cΔ%s\t%sΔ%s\n", <code1>, <code2>, <code3>,
18369 <SID>, <date-time>, <login>

18370 where the entries are:

18371 <code1> A <space> if the delta was applied; '*' otherwise.
18372 <code2> A <space> if the delta was applied or was not applied and ignored; '*' if the delta
18373 was not applied and was not ignored.
18374 <code3> A character indicating a special reason why the delta was or was not applied:
18375 **I** Included.
18376 **X** Excluded.
18377 **C** Cut off (by a **-c** option).
18378 <date-time> Date and time (using the format of the *date* utility's %y/%m/%d %T conversion
18379 specification format) of creation.
18380 <login> Login name of person who created *delta*.

18381 The comments and MR data shall follow on subsequent lines, indented one <tab>. A blank line
18382 shall terminate each entry.

18383 The **p-file** shall be used to pass information resulting from a *get* with a **-e** option along to *delta*.
18384 Its contents shall also be used to prevent a subsequent execution of *get* with a **-e** option for the
18385 same SID until *delta* is executed or the joint edit flag, **j**, is set in the SCCS file. The **p-file** shall be
18386 created in the directory containing the SCCS file and the application shall ensure that the
18387 effective user has write permission in that directory. It shall be writable by owner only, and
18388 owned by the effective user. Each line in the **p-file** shall have the following format:

18389 "%sΔ%sΔ%sΔ%s%s\n", <*g-file SID*>,
18390 <*SID of new delta*>, <*login-name of real user*>,
18391 <*date-time*>, <*i-value*>, <*x-value*>

18392 where <*i-value*> uses the format " " if no **-i** option was specified, and shall use the format:

18393 "Δ-i%s", <*-i option option-argument*>

18394 if a **-i** option was specified and <*x-value*> uses the format " " if no **-x** option was specified, and
18395 shall use the format:

18396 "Δ-x%s", <*-x option option-argument*>

18397 if a **-x** option was specified. There can be an arbitrary number of lines in the **p-file** at any time;
18398 no two lines shall have the same new delta SID.

18399 The **z-file** shall serve as a lock-out mechanism against simultaneous updates. Its contents shall
18400 be the binary process ID of the command (that is, *get*) that created it. The **z-file** shall be created
18401 in the directory containing the SCCS file for the duration of *get*. The same protection restrictions
18402 as those for the **p-file** shall apply for the **z-file**. The **z-file** shall be created read-only.

18403 EXTENDED DESCRIPTION

Determination of SCCS Identification String				
18404	18405	18406	18407	18408
SID* Specified	-b Keyletter Used†	Other Conditions	SID Retrieved	SID of Delta to be Created
none‡	no	R defaults to mR	mR.mL	mR.(mL+1)
none‡	yes	R defaults to mR	mR.mL	mR.mL.(mB+1).1
R	no	R > mR	mR.mL	R.1***
R	no	R = mR	mR.mL	mR.(mL+1)
R	yes	R > mR	mR.mL	mR.mL.(mB+1).1
R	yes	R = mR	mR.mL	mR.mL.(mB+1).1
R	–	R < mR and R does not exist	hR.mL**	hR.mL.(mB+1).1
R	–	Trunk successor in release > R and R exists	R.mL	R.mL.(mB+1).1
R.L	no	No trunk successor	R.L	R.(L+1)
R.L	yes	No trunk successor	R.L	R.L.(mB+1).1
R.L	–	Trunk successor in release ≥ R	R.L	R.L.(mB+1).1
R.L.B	no	No branch successor	R.L.B.mS	R.L.B.(mS+1)
R.L.B	yes	No branch successor	R.L.B.mS	R.L.(mB+1).1
R.L.B.S	no	No branch successor	R.L.B.S	R.L.B.(S+1)
R.L.B.S	yes	No branch successor	R.L.B.S	R.L.(mB+1).1
R.L.B.S	–	Branch successor	R.L.B.S	R.L.(mB+1).1

* R, L, B, and S are the release, level, branch, and sequence components of the SID, respectively; m means maximum. Thus, for example, R.mL means “the maximum level number within release R”; R.L.(mB+1).1 means “the first sequence number on the new branch (that is, maximum branch number plus one) of level L within release R”. Note that if the SID specified is of the form R.L, R.L.B, or R.L.B.S, each of the specified components shall exist.

** hR is the highest existing release that is lower than the specified, nonexistent, release R.

*** This is used to force creation of the first delta in a new release.

† The –b option is effective only if the b flag is present in the file. An entry of ‘–’ means “irrelevant”.

‡ This case applies if the d (default SID) flag is not present in the file. If the d flag is present in the file, then the SID obtained from the d flag is interpreted as if it had been specified on the command line. Thus, one of the other cases in this table applies.

18439	System Date and Time
18440	When a g-file is generated, the creation time of deltas in the SCCS file may be taken into account. If any of these times are apparently in the future, the behavior is unspecified.
18442	Identification Keywords
18443	Identifying information shall be inserted into the text retrieved from the SCCS file by replacing
18444	identification keywords with their value wherever they occur. The following keywords may be
18445	used in the text stored in an SCCS file:
18446	%M% Module name: either the value of the m flag in the file, or if absent, the name of the
18447	SCCS file with the leading s . removed.
18448	%I% SCCS identification (SID) (%R%.%L% or %R%.%L%.%B%.%S%) of the retrieved
18449	text.
18450	%R% Release.
18451	%L% Level.
18452	%B% Branch.
18453	%S% Sequence.
18454	%D% Current date (YY/MM/DD).
18455	%H% Current date (MM/DD/YY).
18456	%T% Current time (HH:MM:SS).
18457	%E% Date newest applied delta was created (YY/MM/DD).
18458	%G% Date newest applied delta was created (MM/DD/YY).
18459	%U% Time newest applied delta was created (HH:MM:SS).
18460	%Y% Module type: value of the t flag in the SCCS file.
18461	%F% SCCS filename.
18462	%P% SCCS absolute pathname.
18463	%Q% The value of the q flag in the file.
18464	%C% Current line number. This keyword is intended for identifying messages output by
18465	the program, such as "this should not have happened" type errors. It is not
18466	intended to be used on every line to provide sequence numbers.
18467	%Z% The four-character string "@(#)" recognizable by <i>what</i> .
18468	%W% A shorthand notation for constructing <i>what</i> strings: % W % = % Z % % M % <tab> % I %
18469	
18470	%A% Another shorthand notation for constructing <i>what</i> strings: % A % = % Z % % Y % % M % % I % % Z %
18471	
18472	EXIT STATUS
18473	The following exit values shall be returned:
18474	0 Successful completion.
18475	>0 An error occurred.

18476 CONSEQUENCES OF ERRORS

18477 Default.

18478 APPLICATION USAGE

18479 Problems can arise if the system date and time have been modified (for example, put forward
18480 and then back again, or unsynchronized clocks across a network) and can also arise when
18481 different values of the *TZ* environment variable are used.

18482 Problems of a similar nature can also arise for the operation of the *delta* utility, which compares
18483 the previous file body against the working file as part of its normal operation.

18484 EXAMPLES

18485 None.

18486 RATIONALE

18487 None.

18488 FUTURE DIRECTIONS

18489 The *-lp* option may be withdrawn in a future version.

18490 SEE ALSO

18491 *admin, delta, prs, what*

18492 CHANGE HISTORY

18493 First released in Issue 2.

18494 Issue 5

18495 A correction is made to the first format string in STDOUT.

18496 The interpretation of the YY component of the *-c cutoff* argument is noted.

18497 Issue 6

18498 The obsolescent SYNOPSIS is removed, removing the *-lp* option.

18499 The normative text is reworded to avoid use of the term “must” for application requirements.

18500 The Open Group Corrigendum U025/5 is applied, correcting text in the OPTIONS section.

18501 The Open Group Corrigendum U048/1 is applied.

18502 The Open Group Interpretation PIN4C.00014 is applied.

18503 The Open Group Base Resolution bwg2001-007 is applied as follows:

- 18504 • The EXTENDED DESCRIPTION section is updated to make partial SID handling
18505 unspecified, reflecting common usage, and to clarify SID ranges.

- 18506 • New text is added to the EXTENDED DESCRIPTION and APPLICATION USAGE sections
18507 regarding how the system date and time may be taken into account.

- 18508 • The *TZ* environment variable is added to the ENVIRONMENT VARIABLES section.

18509 **NAME**

18510 getconf — get configuration values

18511 **SYNOPSIS**

18512 getconf [-v specification] system_var

18513 getconf [-v specification] path_var pathname

18514 **DESCRIPTION**

18515 In the first synopsis form, the *getconf* utility shall write to the standard output the value of the
18516 variable specified by the *system_var* operand.

18517 In the second synopsis form, the *getconf* utility shall write to the standard output the value of the
18518 variable specified by the *path_var* operand for the path specified by the *pathname* operand.

18519 The value of each configuration variable shall be determined as if it were obtained by calling the
18520 function from which it is defined to be available by this volume of IEEE Std 1003.1-2001 or by the
18521 System Interfaces volume of IEEE Std 1003.1-2001 (see the OPERANDS section). The value shall
18522 reflect conditions in the current operating environment.

18523 **OPTIONS**

18524 The *getconf* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
18525 12.2, Utility Syntax Guidelines.

18526 The following option shall be supported:

18527 **-v specification**

18528 Indicate a specific specification and version for which configuration variables shall
18529 be determined. If this option is not specified, the values returned correspond to an
18530 implementation default conforming compilation environment.

18531 If the command:

18532 getconf _POSIX_V6_ILP32_OFF32

18533 does not write "-1\n" or "undefined\n" to standard output, then commands of
18534 the form:

18535 getconf -v POSIX_V6_ILP32_OFF32 ...

18536 determine values for configuration variables corresponding to the
18537 POSIX_V6_ILP32_OFF32 compilation environment specified in c99, the
18538 EXTENDED DESCRIPTION.

18539 If the command:

18540 getconf _POSIX_V6_ILP32_OFFBIG

18541 does not write "-1\n" or "undefined\n" to standard output, then commands of
18542 the form:

18543 getconf -v POSIX_V6_ILP32_OFFBIG ...

18544 determine values for configuration variables corresponding to the
18545 POSIX_V6_ILP32_OFFBIG compilation environment specified in c99, the
18546 EXTENDED DESCRIPTION.

18547 If the command:

18548 getconf _POSIX_V6_LP64_OFF64

18549 does not write "-1\n" or "undefined\n" to standard output, then commands of
18550 the form:

```
18551           getconf -v POSIX_V6_LP64_OFF64 ...
18552           determine values for configuration variables corresponding to the
18553           POSIX_V6_LP64_OFF64 compilation environment specified in c99, the
18554           EXTENDED DESCRIPTION.
18555           If the command:
18556           getconf _POSIX_V6_LPBIG_OFFBIG
18557           does not write "-1\n" or "undefined\n" to standard output, then commands of
18558           the form:
18559           getconf -v POSIX_V6_LPBIG_OFFBIG ...
18560           determine values for configuration variables corresponding to the
18561           POSIX_V6_LPBIG_OFFBIG compilation environment specified in c99, the
18562           EXTENDED DESCRIPTION.
```

18563 OPERANDS

18564 The following operands shall be supported:

18565 <i>path_var</i>	A name of a configuration variable. All of the variables in the <i>pathconf()</i> function 18566 defined in the System Interfaces volume of IEEE Std 1003.1-2001 are supported and 18567 the implementation may add other local variables.
18568 <i>pathname</i>	A pathname for which the variable specified by <i>path_var</i> is to be determined.
18569 <i>system_var</i>	A name of a configuration variable. All of the variables in the <i>confstr()</i> and 18570 <i>sysconf()</i> functions defined in the System Interfaces volume of 18571 IEEE Std 1003.1-2001 shall be supported and the implementation may add other 18572 local values. When the symbol listed in the first column of the following table is used as the 18574 <i>system_var</i> operand, <i>getconf</i> yields the same value as <i>confstr()</i> when called with the 18575 value in the second column:

18576

18577

18578

18579

18580

18581

18582

18583

18584

18585

18586

18587

18588

18589

18590

18591

18592 XSI

18593

18594

18595

18596

18597

18598

18599

18600

18601

18602

18603

18604

18605

18606

18607

	system_var	confstr() Name Value
18578	PATH	CS_PATH
18579	POSIX_V6_ILP32_OFF32_CFLAGS	CS_POSIX_V6_ILP32_OFF32_CFLAGS
18580	POSIX_V6_ILP32_OFF32_LDFLAGS	CS_POSIX_V6_ILP32_OFF32_LDFLAGS
18581	POSIX_V6_ILP32_OFF32_LIBS	CS_POSIX_V6_ILP32_OFF32_LIBS
18582	POSIX_V6_ILP32_OFFBIG_CFLAGS	CS_POSIX_V6_ILP32_OFFBIG_CFLAGS
18583	POSIX_V6_ILP32_OFFBIG_LDFLAGS	CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS
18584	POSIX_V6_ILP32_OFFBIG_LIBS	CS_POSIX_V6_ILP32_OFFBIG_LIBS
18585	POSIX_V6_LP64_OFF64_CFLAGS	CS_POSIX_V6_LP64_OFF64_CFLAGS
18586	POSIX_V6_LP64_OFF64_LDFLAGS	CS_POSIX_V6_LP64_OFF64_LDFLAGS
18587	POSIX_V6_LP64_OFF64_LIBS	CS_POSIX_V6_LP64_OFF64_LIBS
18588	POSIX_V6_LPBIG_OFFBIG_CFLAGS	CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS
18589	POSIX_V6_LPBIG_OFFBIG_LDFLAGS	CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS
18590	POSIX_V6_LPBIG_OFFBIG_LIBS	CS_POSIX_V6_LPBIG_OFFBIG_LIBS
18591	POSIX_V6_WIDTH_RESTRICTED_ENVS	CS_POSIX_V6_WIDTH_RESTRICTED_ENVS
18592 XSI	XBS5_ILP32_OFF32_CFLAGS (LEGACY)	CS_XBS5_ILP32_OFF32_CFLAGS
18593	XBS5_ILP32_OFF32_LDFLAGS (LEGACY)	CS_XBS5_ILP32_OFF32_LDFLAGS
18594	XBS5_ILP32_OFF32_LIBS (LEGACY)	CS_XBS5_ILP32_OFF32_LIBS
18595	XBS5_ILP32_OFF32_LINTFLAGS (LEGACY)	CS_XBS5_ILP32_OFF32_LINTFLAGS
18596	XBS5_ILP32_OFFBIG_CFLAGS (LEGACY)	CS_XBS5_ILP32_OFFBIG_CFLAGS
18597	XBS5_ILP32_OFFBIG_LDFLAGS (LEGACY)	CS_XBS5_ILP32_OFFBIG_LDFLAGS
18598	XBS5_ILP32_OFFBIG_LIBS (LEGACY)	CS_XBS5_ILP32_OFFBIG_LIBS
18599	XBS5_ILP32_OFFBIG_LINTFLAGS (LEGACY)	CS_XBS5_ILPBIG_OFF32_LINTFLAGS
18600	XBS5_LP64_OFF64_CFLAGS (LEGACY)	CS_XBS5_LP64_OFF64_CFLAGS
18601	XBS5_LP64_OFF64_LDFLAGS (LEGACY)	CS_XBS5_LP64_OFF64_LDFLAGS
18602	XBS5_LP64_OFF64_LIBS (LEGACY)	CS_XBS5_LP64_OFF64_LIBS
18603	XBS5_LP64_OFF64_LINTFLAGS (LEGACY)	CS_XBS5_LP64_OFF64_LINTFLAGS
18604	XBS5_LPBIG_OFFBIG_CFLAGS (LEGACY)	CS_XBS5_LPBIG_OFFBIG_CFLAGS
18605	XBS5_LPBIG_OFFBIG_LDFLAGS (LEGACY)	CS_XBS5_LPBIG_OFFBIG_LDFLAGS
18606	XBS5_LPBIG_OFFBIG_LIBS (LEGACY)	CS_XBS5_LPBIG_OFFBIG_LIBS
18607	XBS5_LPBIG_OFFBIG_LINTFLAGS (LEGACY)	CS_XBS5_LPBIG_OFFBIG_LINTFLAGS

18608 **STDIN**

18609 Not used.

18610 **INPUT FILES**

18611 None.

18612 **ENVIRONMENT VARIABLES**18613 The following environment variables shall affect the execution of *getconf*.18614 **LANG** Provide a default value for the internationalization variables that are unset or null.
(See the *Base Definitions* volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)18618 **LC_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.18620 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).18623 **LC_MESSAGES**

18624 Determine the locale that should be used to affect the format and contents of

18625 diagnostic messages written to standard error.

18626 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

18627 **ASYNCHRONOUS EVENTS**

18628 Default.

18629 **STDOUT**

18630 If the specified variable is defined on the system and its value is described to be available from
18631 the *confstr()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001, its value
18632 shall be written in the following format:

18633 "%s\n", <value>

18634 Otherwise, if the specified variable is defined on the system, its value shall be written in the
18635 following format:

18636 "%d\n", <value>

18637 If the specified variable is valid, but is undefined on the system, *getconf* shall write using the
18638 following format:

18639 "undefined\n"

18640 If the variable name is invalid or an error occurs, nothing shall be written to standard output.

18641 **STDERR**

18642 The standard error shall be used only for diagnostic messages.

18643 **OUTPUT FILES**

18644 None.

18645 **EXTENDED DESCRIPTION**

18646 None.

18647 **EXIT STATUS**

18648 The following exit values shall be returned:

18649 0 The specified variable is valid and information about its current state was written
18650 successfully.

18651 >0 An error occurred.

18652 **CONSEQUENCES OF ERRORS**

18653 Default.

18654 **APPLICATION USAGE**

18655 None.

18656 **EXAMPLES**

18657 The following example illustrates the value of {NGROUPS_MAX}:

18658 `getconf NGROUPS_MAX`

18659 The following example illustrates the value of {NAME_MAX} for a specific directory:

18660 `getconf NAME_MAX /usr`

18661 The following example shows how to deal more carefully with results that might be unspecified:

18662 `if value=$(getconf PATH_MAX /usr); then`

18663 `if ["$value" = "undefined"]; then`

18664 `echo PATH_MAX in /usr is infinite.`

18665 `else`

```
18666         echo PATH_MAX in /usr is $value.  
18667     fi  
18668 else  
18669     echo Error in getconf.  
18670 fi  
  
18671 Note that:  
  
18672 sysconf(_SC_POSIX_C_BIND);  
18673 and:  
  
18674 system("getconf POSIX2_C_BIND");  
  
18675 in a C program could give different answers. The sysconf() call supplies a value that corresponds  
18676 to the conditions when the program was either compiled or executed, depending on the  
18677 implementation; the system() call to getconf always supplies a value corresponding to conditions  
18678 when the program is executed.
```

18679 RATIONALE

18680 The original need for this utility, and for the *confstr()* function, was to provide a way of finding
18681 the configuration-defined default value for the *PATH* environment variable. Since *PATH* can be
18682 modified by the user to include directories that could contain utilities replacing the standard
18683 utilities, shell scripts need a way to determine the system-supplied *PATH* environment variable
18684 value that contains the correct search path for the standard utilities. It was later suggested that
18685 access to the other variables described in this volume of IEEE Std 1003.1-2001 could also be
18686 useful to applications.

18687 This functionality of *getconf* would not be adequately subsumed by another command such as:

```
18688 grep var /etc/conf
```

18689 because such a strategy would provide correct values for neither those variables that can vary at
18690 runtime, nor those that can vary depending on the path.

18691 Early proposal versions of *getconf* specified exit status 1 when the specified variable was valid,
18692 but not defined on the system. The output string "undefined" is now used to specify this case
18693 with exit code 0 because so many things depend on an exit code of zero when an invoked utility
18694 is successful.

18695 FUTURE DIRECTIONS

18696 None.

18697 SEE ALSO

18698 *c99*, the System Interfaces volume of IEEE Std 1003.1-2001, *confstr()*, *pathconf()*, *sysconf()*,
18699 *system()*

18700 CHANGE HISTORY

18701 First released in Issue 4.

18702 Issue 5

18703 In the OPERANDS section:

- 18704 • {NL_MAX} is changed to {NL_NMAX}.
- 18705 • Entries beginning NL_ are deleted from the list of standard configuration variables.
- 18706 • The list of variables previously marked UX is merged with the list marked EX.
- 18707 • Operands are added to support new Option Groups.

- 18708 • Operands are added so that *getconf* can determine supported programming environments.

18709 Issue 6

18710 The Open Group Corrigendum U029/4 is applied, correcting the example command in the last
18711 paragraph of the OPTIONS section.

18712 The following new requirements on POSIX implementations derive from alignment with the
18713 Single UNIX Specification:

- 18714 • Operands are added to determine supported programming environments.

18715 This reference page is updated for alignment with the ISO/IEC 9899:1999 standard. Specifically,
18716 new macros for *c99* programming environments are introduced.

18717 XSI marked *system_var* (XBS5_*) values are marked LEGACY.

18718 NAME

18719 getopts — parse utility options

18720 SYNOPSIS

18721 getopts *optstring name [arg...]*

18722 DESCRIPTION

18723 The *getopts* utility shall retrieve options and option-arguments from a list of parameters. It shall
18724 support the Utility Syntax Guidelines 3 to 10, inclusive, described in the Base Definitions volume
18725 of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

18726 Each time it is invoked, the *getopts* utility shall place the value of the next option in the shell
18727 variable specified by the *name* operand and the index of the next argument to be processed in the
18728 shell variable *OPTIND*. Whenever the shell is invoked, *OPTIND* shall be initialized to 1.

18729 When the option requires an option-argument, the *getopts* utility shall place it in the shell
18730 variable *OPTARG*. If no option was found, or if the option that was found does not have an
18731 option-argument, *OPTARG* shall be unset.

18732 If an option character not contained in the *optstring* operand is found where an option character
18733 is expected, the shell variable specified by *name* shall be set to the question-mark ('?') character.
18734 In this case, if the first character in *optstring* is a colon (':'), the shell variable *OPTARG* shall be
18735 set to the option character found, but no output shall be written to standard error; otherwise, the
18736 shell variable *OPTARG* shall be unset and a diagnostic message shall be written to standard
18737 error. This condition shall be considered to be an error detected in the way arguments were
18738 presented to the invoking application, but shall not be an error in *getopts* processing.

18739 If an option-argument is missing:

- If the first character of *optstring* is a colon, the shell variable specified by *name* shall be set to
the colon character and the shell variable *OPTARG* shall be set to the option character found.
- Otherwise, the shell variable specified by *name* shall be set to the question-mark character,
the shell variable *OPTARG* shall be unset, and a diagnostic message shall be written to
standard error. This condition shall be considered to be an error detected in the way
arguments were presented to the invoking application, but shall not be an error in *getopts*
processing; a diagnostic message shall be written as stated, but the exit status shall be zero.

18747 When the end of options is encountered, the *getopts* utility shall exit with a return value greater
18748 than zero; the shell variable *OPTIND* shall be set to the index of the first non-option-argument,
18749 where the first "--" argument is considered to be an option-argument if there are no other
18750 non-option-arguments appearing before it, or the value "\$#+1 if there are no non-option-
18751 arguments; the *name* variable shall be set to the question-mark character. Any of the following
18752 shall identify the end of options: the special option "--", finding an argument that does not
18753 begin with a '--', or encountering an error.

18754 The shell variables *OPTIND* and *OPTARG* shall be local to the caller of *getopts* and shall not be
18755 exported by default.

18756 The shell variable specified by the *name* operand, *OPTIND*, and *OPTARG* shall affect the current
18757 shell execution environment; see Section 2.12 (on page 61).

18758 If the application sets *OPTIND* to the value 1, a new set of parameters can be used: either the
18759 current positional parameters or new *arg* values. Any other attempt to invoke *getopts* multiple
18760 times in a single shell execution environment with parameters (positional parameters or *arg*
18761 operands) that are not the same in all invocations, or with an *OPTIND* value modified to be a
18762 value other than 1, produces unspecified results.

18763 **OPTIONS**

18764 None.

18765 **OPERANDS**

18766 The following operands shall be supported:

- 18767 *optstring* A string containing the option characters recognized by the utility invoking *getopts*.
 18768 If a character is followed by a colon, the option shall be expected to have an
 18769 argument, which should be supplied as a separate argument. Applications should
 18770 specify an option character and its option-argument as separate arguments, but
 18771 *getopts* shall interpret the characters following an option character requiring
 18772 arguments as an argument whether or not this is done. An explicit null option-
 18773 argument need not be recognized if it is not supplied as a separate argument when
 18774 *getopts* is invoked. (See also the *getopt()* function defined in the System Interfaces
 18775 volume of IEEE Std 1003.1-2001.) The characters question-mark and colon shall not
 18776 be used as option characters by an application. The use of other option characters
 18777 that are not alphanumeric produces unspecified results. If the option-argument is
 18778 not supplied as a separate argument from the option character, the value in
 18779 *OPTARG* shall be stripped of the option character and the '-'. The first character
 18780 in *optstring* determines how *getopts* behaves if an option character is not known or
 18781 an option-argument is missing.
- 18782 *name* The name of a shell variable that shall be set by the *getopts* utility to the option
 18783 character that was found.
- 18784 The *getopts* utility by default shall parse positional parameters passed to the invoking shell
 18785 procedure. If *args* are given, they shall be parsed instead of the positional parameters.

18786 **STDIN**

18787 Not used.

18788 **INPUT FILES**

18789 None.

18790 **ENVIRONMENT VARIABLES**18791 The following environment variables shall affect the execution of *getopts*:

- 18792 *LANG* Provide a default value for the internationalization variables that are unset or null.
 18793 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
 18794 Internationalization Variables for the precedence of internationalization variables
 18795 used to determine the values of locale categories.)
- 18796 *LC_ALL* If set to a non-empty string value, override the values of all the other
 18797 internationalization variables.
- 18798 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
 18799 characters (for example, single-byte as opposed to multi-byte characters in
 18800 arguments and input files).
- 18801 *LC_MESSAGES*
 18802 Determine the locale that should be used to affect the format and contents of
 18803 diagnostic messages written to standard error.
- 18804 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.
- 18805 *OPTIND* This variable shall be used by the *getopts* utility as the index of the next argument
 18806 to be processed.

18807 ASYNCHRONOUS EVENTS

18808 Default.

18809 STDOUT

18810 Not used.

18811 STDERR

18812 Whenever an error is detected and the first character in the *optstring* operand is not a colon (' : '), a diagnostic message shall be written to standard error with the following information in an unspecified format:

- The invoking program name shall be identified in the message. The invoking program name shall be the value of the shell special parameter 0 (see Section 2.5.2 (on page 34)) at the time the *getopts* utility is invoked. A name equivalent to:

18818 basename "\$0"

18819 may be used.

- If an option is found that was not specified in *optstring*, this error is identified and the invalid option character shall be identified in the message.
- If an option requiring an option-argument is found, but an option-argument is not found, this error shall be identified and the invalid option character shall be identified in the message.

18825 OUTPUT FILES

18826 None.

18827 EXTENDED DESCRIPTION

18828 None.

18829 EXIT STATUS

18830 The following exit values shall be returned:

18831 0 An option, specified or unspecified by *optstring*, was found.

18832 >0 The end of options was encountered or an error occurred.

18833 CONSEQUENCES OF ERRORS

18834 Default.

18835 APPLICATION USAGE

18836 Since *getopts* affects the current shell execution environment, it is generally provided as a shell regular built-in. If it is called in a subshell or separate utility execution environment, such as one of the following:

18839 (*getopts abc value "\$@"*)
18840 nohup *getopts* ...
18841 find . -exec *getopts* ... \;

18842 it does not affect the shell variables in the caller's environment.

18843 Note that shell functions share *OPTIND* with the calling shell even though the positional parameters are changed. If the calling shell and any of its functions uses *getopts* to parse arguments, the results are unspecified.

18846 EXAMPLES

18847 The following example script parses and displays its arguments:

18848 *aflag=*
18849 *bflag=*

```

18850     while getopts ab: name
18851     do
18852         case $name in
18853             a)      aflag=1;;
18854             b)      bflag=1
18855                 bval="$OPTARG";;
18856             ?)      printf "Usage: %s: [-a] [-b value] args\n" $0
18857                 exit 2;;
18858         esac
18859     done
18860     if [ ! -z "$aflag" ]; then
18861         printf "Option -a specified\n"
18862     fi
18863     if [ ! -z "$bflag" ]; then
18864         printf 'Option -b "%s" specified\n' "$bval"
18865     fi
18866     shift $((OPTIND - 1))
18867     printf "Remaining arguments are: %s\n" "$*"

```

18868 RATIONALE

18869 The *getopts* utility was chosen in preference to the System V *getopt* utility because *getopts* handles
 18870 option-arguments containing <blank>s.

18871 The *OPTARG* variable is not mentioned in the ENVIRONMENT VARIABLES section because it
 18872 does not affect the execution of *getopts*; it is one of the few “output-only” variables used by the
 18873 standard utilities.

18874 The colon is not allowed as an option character because that is not historical behavior, and it
 18875 violates the Utility Syntax Guidelines. The colon is now specified to behave as in the KornShell
 18876 version of the *getopts* utility; when used as the first character in the *optstring* operand, it disables
 18877 diagnostics concerning missing option-arguments and unexpected option characters. This
 18878 replaces the use of the *OPTERR* variable that was specified in an early proposal.

18879 The formats of the diagnostic messages produced by the *getopts* utility and the *getopt()* function
 18880 are not fully specified because implementations with superior (“friendlier”) formats objected to
 18881 the formats used by some historical implementations. The standard developers considered it
 18882 important that the information in the messages used be uniform between *getopts* and *getopt()*.
 18883 Exact duplication of the messages might not be possible, particularly if a utility is built on
 18884 another system that has a different *getopt()* function, but the messages must have specific
 18885 information included so that the program name, invalid option character, and type of error can
 18886 be distinguished by a user.

18887 Only a rare application program intercepts a *getopts* standard error message and wants to parse
 18888 it. Therefore, implementations are free to choose the most usable messages they can devise. The
 18889 following formats are used by many historical implementations:

```

18890 "%s: illegal option -- %c\n", <program name>, <option character>
18891 "%s: option requires an argument -- %c\n", <program name>, \
18892     <option character>

```

18893 Historical shells with built-in versions of *getopt()* or *getopts* have used different formats,
 18894 frequently not even indicating the option character found in error.

18895 FUTURE DIRECTIONS

18896 None.

18897 SEE ALSO

18898 Section 2.5.2 (on page 34), the System Interfaces volume of IEEE Std 1003.1-2001, *getopt()*

18899 CHANGE HISTORY

18900 First released in Issue 4.

18901 Issue 6

18902 The normative text is reworded to avoid use of the term “must” for application requirements.

18903 NAME

18904 grep — search a file for a pattern

18905 SYNOPSIS

```
18906     grep [-E| -F][-c| -l| -q][-insvx] -e pattern_list...
18907     [-f pattern_file]...[file...]
18908     grep [-E| -F][-c| -l| -q][-insvx][-e pattern_list]...
18909     -f pattern_file...[file...]
18910     grep [-E| -F][-c| -l| -q][-insvx] pattern_list[file...]
```

18911 DESCRIPTION

18912 The *grep* utility shall search the input files, selecting lines matching one or more patterns; the
18913 types of patterns are controlled by the options specified. The patterns are specified by the **-e**
18914 option, **-f** option, or the *pattern_list* operand. The *pattern_list*'s value shall consist of one or more
18915 patterns separated by <newline>s; the *pattern_file*'s contents shall consist of one or more
18916 patterns terminated by <newline>. By default, an input line shall be selected if any pattern,
18917 treated as an entire basic regular expression (BRE) as described in the Base Definitions volume of
18918 IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions, matches any part of the line
18919 excluding the terminating <newline>; a null BRE shall match every line. By default, each selected
18920 input line shall be written to the standard output.

18921 Regular expression matching shall be based on text lines. Since a <newline> separates or
18922 terminates patterns (see the **-e** and **-f** options below), regular expressions cannot contain a
18923 <newline>. Similarly, since patterns are matched against individual lines (excluding the
18924 terminating <newline>s) of the input, there is no way for a pattern to match a <newline> found
18925 in the input.

18926 OPTIONS

18927 The *grep* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
18928 12.2, Utility Syntax Guidelines.

18929 The following options shall be supported:

18930 **-E** Match using extended regular expressions. Treat each pattern specified as an ERE,
18931 as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.4,
18932 Extended Regular Expressions. If any entire ERE pattern matches some part of an
18933 input line excluding the terminating <newline>, the line shall be matched. A null
18934 ERE shall match every line.

18935 **-F** Match using fixed strings. Treat each pattern specified as a string instead of a
18936 regular expression. If an input line contains any of the patterns as a contiguous
18937 sequence of bytes, the line shall be matched. A null string shall match every line.

18938 **-c** Write only a count of selected lines to standard output.

18939 **-e** *pattern_list* Specify one or more patterns to be used during the search for input. The
18940 application shall ensure that patterns in *pattern_list* are separated by a <newline>. The
18941 application shall ensure that patterns in *pattern_list* are separated by a <newline>. A null pattern
18942 can be specified by two adjacent <newline>s in *pattern_list*. Unless
18943 the **-E** or **-F** option is also specified, each pattern shall be treated as a BRE, as
18944 described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic
18945 Regular Expressions. Multiple **-e** and **-f** options shall be accepted by the *grep*
18946 utility. All of the specified patterns shall be used when matching lines, but the
18947 order of evaluation is unspecified.

18948	-f <i>pattern_file</i>	Read one or more patterns from the file named by the pathname <i>pattern_file</i> . Patterns in <i>pattern_file</i> shall be terminated by a <newline>. A null pattern can be specified by an empty line in <i>pattern_file</i> . Unless the -E or -F option is also specified, each pattern shall be treated as a BRE, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions.
18954	-i	Perform pattern matching in searches without regard to case; see the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.2, Regular Expression General Requirements.
18957	-l	(The letter ell.) Write only the names of files containing selected lines to standard output. Pathnames shall be written once per file searched. If the standard input is searched, a pathname of "(standard input)" shall be written, in the POSIX locale. In other locales, "standard input" may be replaced by something more appropriate in those locales.
18962	-n	Precede each output line by its relative line number in the file, each file starting at line 1. The line number counter shall be reset for each file processed.
18964	-q	Quiet. Nothing shall be written to the standard output, regardless of matching lines. Exit with zero status if an input line is selected.
18966	-s	Suppress the error messages ordinarily written for nonexistent or unreadable files. Other error messages shall not be suppressed.
18968	-v	Select lines not matching any of the specified patterns. If the -v option is not specified, selected lines shall be those that match any of the specified patterns.
18970	-x	Consider only input lines that use all characters in the line excluding the terminating <newline> to match an entire fixed string or regular expression to be matching lines.

18973 OPERANDS

18974 The following operands shall be supported:

18975	<i>pattern_list</i>	Specify one or more patterns to be used during the search for input. This operand shall be treated as if it were specified as -e <i>pattern_list</i> .
18977	<i>file</i>	A pathname of a file to be searched for the patterns. If no <i>file</i> operands are specified, the standard input shall be used.

18979 STDIN

18980 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES section.

18982 INPUT FILES

18983 The input files shall be text files.

18984 ENVIRONMENT VARIABLES

18985 The following environment variables shall affect the execution of *grep*:

18986	LANG	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
18990	LC_ALL	If set to a non-empty string value, override the values of all the other internationalization variables.

18992 ***LC_COLLATE***
18993 Determine the locale for the behavior of ranges, equivalence classes, and multi-
18994 character collating elements within regular expressions.

18995 ***LC_CTYPE*** Determine the locale for the interpretation of sequences of bytes of text data as
18996 characters (for example, single-byte as opposed to multi-byte characters in
18997 arguments and input files) and the behavior of character classes within regular
18998 expressions.

18999 ***LC_MESSAGES***
19000 Determine the locale that should be used to affect the format and contents of
19001 diagnostic messages written to standard error.

19002 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

19003 **ASYNCHRONOUS EVENTS**
19004 Default.

19005 **STDOUT**
19006 If the **-l** option is in effect, and the **-q** option is not, the following shall be written for each file
19007 containing at least one selected input line:
19008 "%s\n", <file>
19009 Otherwise, if more than one *file* argument appears, and **-q** is not specified, the *grep* utility shall
19010 prefix each output line by:
19011 "%s:", <file>
19012 The remainder of each output line shall depend on the other options specified:
19013 • If the **-c** option is in effect, the remainder of each output line shall contain:
19014 "%d\n", <count>
19015 • Otherwise, if **-c** is not in effect and the **-n** option is in effect, the following shall be written to
19016 standard output:
19017 "%d:", <line number>
19018 • Finally, the following shall be written to standard output:
19019 "%s", <selected-line contents>

19020 **STDERR**
19021 The standard error shall be used only for diagnostic messages.

19022 **OUTPUT FILES**
19023 None.

19024 **EXTENDED DESCRIPTION**
19025 None.

19026 **EXIT STATUS**
19027 The following exit values shall be returned:
19028 0 One or more lines were selected.
19029 1 No lines were selected.
19030 >1 An error occurred.

19031 CONSEQUENCES OF ERRORS

19032 If the **-q** option is specified, the exit status shall be zero if an input line is selected, even if an
19033 error was detected. Otherwise, default actions shall be performed.

19034 APPLICATION USAGE

19035 Care should be taken when using characters in *pattern_list* that may also be meaningful to the
19036 command interpreter. It is safest to enclose the entire *pattern_list* argument in single quotes:

19037 ' . . . '

19038 The **-e** *pattern_list* option has the same effect as the *pattern_list* operand, but is useful when
19039 *pattern_list* begins with the hyphen delimiter. It is also useful when it is more convenient to
19040 provide multiple patterns as separate arguments.

19041 Multiple **-e** and **-f** options are accepted and *grep* uses all of the patterns it is given while
19042 matching input text lines. (Note that the order of evaluation is not specified. If an
19043 implementation finds a null string as a pattern, it is allowed to use that pattern first, matching
19044 every line, and effectively ignore any other patterns.)

19045 The **-q** option provides a means of easily determining whether or not a pattern (or string) exists
19046 in a group of files. When searching several files, it provides a performance improvement
19047 (because it can quit as soon as it finds the first match) and requires less care by the user in
19048 choosing the set of files to supply as arguments (because it exits zero if it finds a match even if
19049 *grep* detected an access or read error on earlier *file* operands).

19050 EXAMPLES

- 19051 1. To find all uses of the word "Posix" (in any case) in file **text.mm** and write with line
19052 numbers:

19053 **grep -i -n posix text.mm**

- 19054 2. To find all empty lines in the standard input:

19055 **grep ^\$**

19056 or:

19057 **grep -v .**

- 19058 3. Both of the following commands print all lines containing strings "abc" or "def" or both:

19059 **grep -E 'abc|def'**

19060 **grep -F 'abc|def'**

- 19061 4. Both of the following commands print all lines matching exactly "abc" or "def":

19062 **grep -E '^abc\$|^def\$'**

19063 **grep -F -x 'abc|def'**

19064 RATIONALE

19065 This *grep* has been enhanced in an upwards-compatible way to provide the exact functionality of
19066 the historical *egrep* and *fgrep* commands as well. It was the clear intention of the standard
19067 developers to consolidate the three *greps* into a single command.

19068 The old *egrep* and *fgrep* commands are likely to be supported for many years to come as
19069 implementation extensions, allowing historical applications to operate unmodified.

19070 Historical implementations usually silently ignored all but one of multiply-specified **-e** and **-f**
19071 options, but were not consistent as to which specification was actually used.

19072 The **-b** option was omitted from the OPTIONS section because block numbers are
19073 implementation-defined.

19074 The System V restriction on using – to mean standard input was omitted.

19075 A definition of action taken when given a null BRE or ERE is specified. This is an error condition
19076 in some historical implementations.

19077 The **-l** option previously indicated that its use was undefined when no files were explicitly
19078 named. This behavior was historical and placed an unnecessary restriction on future
19079 implementations. It has been removed.

19080 The historical BSD *grep -s* option practice is easily duplicated by redirecting standard output to
19081 **/dev/null**. The **-s** option required here is from System V.

19082 The **-x** option, historically available only with *fgrep*, is available here for all of the non-
19083 obsolescent versions.

19084 FUTURE DIRECTIONS

19085 None.

19086 SEE ALSO

19087 *sed*

19088 CHANGE HISTORY

19089 First released in Issue 2.

19090 Issue 6

19091 The Open Group Corrigendum U029/5 is applied, correcting the SYNOPSIS.

19092 The normative text is reworded to avoid use of the term “must” for application requirements.

19093 NAME

19094 hash — remember or report utility locations

19095 SYNOPSIS

19096 XSI hash [*utility...*]

19097 hash -r

19098

19099 DESCRIPTION

19100 The *hash* utility shall affect the way the current shell environment remembers the locations of
19101 utilities found as described in Section 2.9.1.1 (on page 48). Depending on the arguments
19102 specified, it shall add utility locations to its list of remembered locations or it shall purge the
19103 contents of the list. When no arguments are specified, it shall report on the contents of the list.

19104 Utilities provided as built-ins to the shell shall not be reported by *hash*.

19105 OPTIONS

19106 The *hash* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
19107 12.2, Utility Syntax Guidelines.

19108 The following option shall be supported:

19109 -r Forget all previously remembered utility locations.

19110 OPERANDS

19111 The following operand shall be supported:

19112 *utility* The name of a utility to be searched for and added to the list of remembered
19113 locations. If *utility* contains one or more slashes, the results are unspecified.

19114 STDIN

19115 Not used.

19116 INPUT FILES

19117 None.

19118 ENVIRONMENT VARIABLES

19119 The following environment variables shall affect the execution of *hash*:

19120 *LANG* Provide a default value for the internationalization variables that are unset or null.
19121 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19122 Internationalization Variables for the precedence of internationalization variables
19123 used to determine the values of locale categories.)

19124 *LC_ALL* If set to a non-empty string value, override the values of all the other
19125 internationalization variables.

19126 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
19127 characters (for example, single-byte as opposed to multi-byte characters in
19128 arguments).

19129 *LC_MESSAGES*

19130 Determine the locale that should be used to affect the format and contents of
19131 diagnostic messages written to standard error.

19132 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

19133 *PATH* Determine the location of *utility*, as described in the Base Definitions volume of
19134 IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

19135 ASYNCHRONOUS EVENTS

19136 Default.

19137 STDOUT

19138 The standard output of *hash* shall be used when no arguments are specified. Its format is
19139 unspecified, but includes the pathname of each utility in the list of remembered locations for the
19140 current shell environment. This list shall consist of those utilities named in previous *hash*
19141 invocations that have been invoked, and may contain those invoked and found through the
19142 normal command search process.

19143 STDERR

19144 The standard error shall be used only for diagnostic messages.

19145 OUTPUT FILES

19146 None.

19147 EXTENDED DESCRIPTION

19148 None.

19149 EXIT STATUS

19150 The following exit values shall be returned:

19151 0 Successful completion.

19152 >0 An error occurred.

19153 CONSEQUENCES OF ERRORS

19154 Default.

19155 APPLICATION USAGE

19156 Since *hash* affects the current shell execution environment, it is always provided as a shell
19157 regular built-in. If it is called in a separate utility execution environment, such as one of the
19158 following:

19159 nohup hash -r
19160 find . -type f | xargs hash

19161 it does not affect the command search process of the caller's environment.

19162 The *hash* utility may be implemented as an alias—for example, *alias -t -*, in which case utilities
19163 found through normal command search are not listed by the *hash* command.

19164 The effects of *hash -r* can also be achieved portably by resetting the value of *PATH*; in the
19165 simplest form, this can be:

19166 PATH= "\$PATH"

19167 The use of *hash* with *utility* names is unnecessary for most applications, but may provide a
19168 performance improvement on a few implementations; normally, the hashing process is included
19169 by default.

19170 EXAMPLES

19171 None.

19172 RATIONALE

19173 None.

19174 FUTURE DIRECTIONS

19175 None.

19176 SEE ALSO

19177 Section 2.9.1.1 (on page 48)

19178 CHANGE HISTORY

19179 First released in Issue 2.

19180 NAME

19181 head — copy the first part of files

19182 SYNOPSIS

19183 head [-n *number*] [*file...*]

19184 DESCRIPTION

19185 The *head* utility shall copy its input files to the standard output, ending the output for each file at
19186 a designated point.

19187 Copying shall end at the point in each input file indicated by the **-n** *number* option. The option-
19188 argument *number* shall be counted in units of lines.

19189 OPTIONS

19190 The *head* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
19191 12.2, Utility Syntax Guidelines.

19192 The following option shall be supported:

19193 **-n** *number* The first *number* lines of each input file shall be copied to standard output. The
19194 application shall ensure that the *number* option-argument is a positive decimal
19195 integer.

19196 When a file contains less than *number* lines, it shall be copied to standard output in its entirety.
19197 This shall not be an error.

19198 If no options are specified, *head* shall act as if **-n 10** had been specified.

19199 OPERANDS

19200 The following operand shall be supported:

19201 *file* A pathname of an input file. If no *file* operands are specified, the standard input
19202 shall be used.

19203 STDIN

19204 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES
19205 section.

19206 INPUT FILES

19207 Input files shall be text files, but the line length is not restricted to {LINE_MAX} bytes.

19208 ENVIRONMENT VARIABLES

19209 The following environment variables shall affect the execution of *head*:

19210 *LANG* Provide a default value for the internationalization variables that are unset or null.
19211 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19212 Internationalization Variables for the precedence of internationalization variables
19213 used to determine the values of locale categories.)

19214 *LC_ALL* If set to a non-empty string value, override the values of all the other
19215 internationalization variables.

19216 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
19217 characters (for example, single-byte as opposed to multi-byte characters in
19218 arguments and input files).

19219 *LC_MESSAGES*

19220 Determine the locale that should be used to affect the format and contents of
19221 diagnostic messages written to standard error.

19222 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

19223 **ASYNCHRONOUS EVENTS**

19224 Default.

19225 **STDOUT**

19226 The standard output shall contain designated portions of the input files.

19227 If multiple *file* operands are specified, *head* shall precede the output for each with the header:

19228 "*\n==> %s <=\n*", *<pathname>*

19229 except that the first header written shall not include the initial <newline>.

19230 **STDERR**

19231 The standard error shall be used only for diagnostic messages.

19232 **OUTPUT FILES**

19233 None.

19234 **EXTENDED DESCRIPTION**

19235 None.

19236 **EXIT STATUS**

19237 The following exit values shall be returned:

19238 0 Successful completion.

19239 >0 An error occurred.

19240 **CONSEQUENCES OF ERRORS**

19241 Default.

19242 **APPLICATION USAGE**

19243 The obsolescent *-number* form is withdrawn in this version. Applications should use the *-n* option.

19245 **EXAMPLES**

19246 To write the first ten lines of all files (except those with a leading period) in the directory:

19247 *head **

19248 **RATIONALE**

19249 Although it is possible to simulate *head* with *sed 10q* for a single file, the standard developers decided that the popularity of *head* on historical BSD systems warranted its inclusion alongside *tail*.

19252 This standard version of *head* follows the Utility Syntax Guidelines. The *-n* option was added to this new interface so that *head* and *tail* would be more logically related.

19254 There is no *-c* option (as there is in *tail*) because it is not historical practice and because other utilities in this volume of IEEE Std 1003.1-2001 provide similar functionality.

19256 **FUTURE DIRECTIONS**

19257 None.

19258 **SEE ALSO**

19259 *sed, tail*

19260 CHANGE HISTORY

19261 First released in Issue 4.

19262 Issue 6

19263 The obsolescent **-number** form is withdrawn.

19264 The normative text is reworded to avoid use of the term “must” for application requirements.

19265 The DESCRIPTION is updated to clarify that when a file contains less than the number of lines requested, the entire file is copied to standard output.
19266

19267 NAME

19268 iconv — codeset conversion

19269 SYNOPSIS

19270 iconv [-cs] -f *fromcode* -t *tocode* [*file* ...]

19271 iconv -l

19272 DESCRIPTION

19273 The *iconv* utility shall convert the encoding of characters in *file* from one codeset to another and
19274 write the results to standard output.

19275 When the options indicate that charmap files are used to specify the codesets (see OPTIONS),
19276 the codeset conversion shall be accomplished by performing a logical join on the symbolic
19277 character names in the two charmaps. The implementation need not support the use of charmap
19278 files for codeset conversion unless the POSIX2_LOCALEDEF symbol is defined on the system.

19279 OPTIONS

19280 The *iconv* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
19281 12.2, Utility Syntax Guidelines.

19282 The following options shall be supported:

19283 **-c** Omit any invalid characters from the output. When **-c** is not used, the results of
19284 encountering invalid characters in the input stream (either those that are not valid
19285 members of the *fromcode* or those that have no corresponding value in *tocode*) shall
19286 be specified in the system documentation. The presence or absence of **-c** shall not
19287 affect the exit status of *iconv*.

19288 **-f *fromcode*** Identify the codeset of the input file. If the option-argument contains a slash
19289 character, *iconv* shall attempt to use it as the pathname of a charmap file, as
19290 defined in the Base Definitions volume of IEEE Std 1003.1-2001, Section 6.4,
19291 Character Set Description File. If the pathname does not represent a valid, readable
19292 charmap file, the results are undefined. If the option-argument does not contain a
19293 slash, it shall be considered the name of one of the codeset descriptions provided
19294 by the system, in an unspecified format. The valid values of the option-argument
19295 without a slash are implementation-defined. If this option is omitted, the codeset
19296 of the current locale shall be used.

19297 **-l** Write all supported *fromcode* and *tocode* values to standard output in an unspecified
19298 format.

19299 **-s** Suppress any messages written to standard error concerning invalid characters.
19300 When **-s** is not used, the results of encountering invalid characters in the input
19301 stream (either those that are not valid members of the *fromcode* or those that have
19302 no corresponding value in *tocode*) shall be specified in the system documentation.
19303 The presence or absence of **-s** shall not affect the exit status of *iconv*.

19304 **-t *tocode*** Identify the codeset to be used for the output file. The semantics shall be
19305 equivalent to the **-f *fromcode*** option.

19306 If either **-f** or **-t** represents a charmap file, but the other does not (or is omitted), or both **-f** and
19307 **-t** are omitted, the results are undefined.

19308 OPERANDS

19309 The following operand shall be supported:

19310 **file** A pathname of an input file. If no *file* operands are specified, or if a *file* operand is
19311 '**-**', the standard input shall be used.

19312 STDIN

19313 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-'.

19314 INPUT FILES

19315 The input file shall be a text file.

19316 ENVIRONMENT VARIABLES

19317 The following environment variables shall affect the execution of *iconv*:

19318 *LANG* Provide a default value for the internationalization variables that are unset or null.
19319 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19320 Internationalization Variables for the precedence of internationalization variables
19321 used to determine the values of locale categories.)

19322 *LC_ALL* If set to a non-empty string value, override the values of all the other
19323 internationalization variables.

19324 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
19325 characters (for example, single-byte as opposed to multi-byte characters in
19326 arguments). During translation of the file, this variable is superseded by the use of
19327 the *fromcode* option-argument.

19328 *LC_MESSAGES*

19329 Determine the locale that should be used to affect the format and contents of
19330 diagnostic messages written to standard error.

19331 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

19332 ASYNCHRONOUS EVENTS

19333 Default.

19334 STDOUT

19335 When the **-l** option is used, the standard output shall contain all supported *fromcode* and *tocode*
19336 values, written in an unspecified format.

19337 When the **-l** option is not used, the standard output shall contain the sequence of characters
19338 read from the input files, translated to the specified codeset. Nothing else shall be written to the
19339 standard output.

19340 STDERR

19341 The standard error shall be used only for diagnostic messages.

19342 OUTPUT FILES

19343 None.

19344 EXTENDED DESCRIPTION

19345 None.

19346 EXIT STATUS

19347 The following exit values shall be returned:

19348 0 Successful completion.

19349 >0 An error occurred.

19350 CONSEQUENCES OF ERRORS

19351 Default.

19352 APPLICATION USAGE

19353 The user must ensure that both charmap files use the same symbolic names for characters the
19354 two codesets have in common.

19355 EXAMPLES

19356 The following example converts the contents of file **mail.x400** from the ISO/IEC 6937:1994
19357 standard codeset to the ISO/IEC 8859-1:1998 standard codeset, and stores the results in file
19358 **mail.local**:

19359 `iconv -f IS6937 -t IS8859 mail.x400 > mail.local`

19360 RATIONALE

19361 The **iconv** utility can be used portably only when the user provides two charmap files as option-
19362 arguments. This is because a single charmap provided by the user cannot reliably be joined with
19363 the names in a system-provided character set description. The valid values for *fromcode* and
19364 *tocode* are implementation-defined and do not have to have any relation to the charmap
19365 mechanisms. As an aid to interactive users, the **-I** option was adopted from the Plan 9 operating
19366 system. It writes information concerning these implementation-defined values. The format is
19367 unspecified because there are many possible useful formats that could be chosen, such as a
19368 matrix of valid combinations of *fromcode* and *tocode*. The **-I** option is not intended for shell script
19369 usage; conforming applications will have to use charmaps.

19370 FUTURE DIRECTIONS

19371 None.

19372 SEE ALSO

19373 *gencat*

19374 CHANGE HISTORY

19375 First released in Issue 3.

19376 Issue 6

19377 This utility has been rewritten to align with the IEEE P1003.2b draft standard. Specifically, the
19378 ability to use charmap files for conversion has been added.

19379 **NAME**

19380 id — return user identity

19381 **SYNOPSIS**

19382 id [user]

19383 id -G[-n] [user]

19384 id -g[-nr] [user]

19385 id -u[-nr] [user]

19386 **DESCRIPTION**

19387 If no *user* operand is provided, the *id* utility shall write the user and group IDs and the corresponding user and group names of the invoking process to standard output. If the effective and real IDs do not match, both shall be written. If multiple groups are supported by the underlying system (see the description of {NGROUPS_MAX} in the System Interfaces volume of IEEE Std 1003.1-2001), the supplementary group affiliations of the invoking process shall also be written.

19393 If a *user* operand is provided and the process has the appropriate privileges, the user and group IDs of the selected user shall be written. In this case, effective IDs shall be assumed to be identical to real IDs. If the selected user has more than one allowable group membership listed in the group database, these shall be written in the same manner as the supplementary groups described in the preceding paragraph.

19398 **OPTIONS**

19399 The *id* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

19401 The following options shall be supported:

19402 **-G** Output all different group IDs (effective, real, and supplementary) only, using the format "%u\n". If there is more than one distinct group affiliation, output each such affiliation, using the format "%u", before the <newline> is output.

19405 **-g** Output only the effective group ID, using the format "%u\n".

19406 **-n** Output the name in the format "%s" instead of the numeric ID using the format "%u".

19408 **-r** Output the real ID instead of the effective ID.

19409 **-u** Output only the effective user ID, using the format "%u\n".

19410 **OPERANDS**

19411 The following operand shall be supported:

19412 *user* The login name for which information is to be written.

19413 **STDIN**

19414 Not used.

19415 **INPUT FILES**

19416 None.

19417 **ENVIRONMENT VARIABLES**

19418 The following environment variables shall affect the execution of *id*:

19419 **LANG** Provide a default value for the internationalization variables that are unset or null.
19420 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19421 Internationalization Variables for the precedence of internationalization variables

19422	used to determine the values of locale categories.)
19423	<i>LC_ALL</i> If set to a non-empty string value, override the values of all the other internationalization variables.
19424	
19425	<i>LC_CTYPE</i> Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
19426	
19427	
19428	<i>LC_MESSAGES</i>
19429	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.
19430	
19431	
19432 XSI	<i>NLSPATH</i> Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
19433	ASYNCHRONOUS EVENTS
19434	Default.
19435	STDOUT
19436	The following formats shall be used when the <i>LC_MESSAGES</i> locale category specifies the POSIX locale. In other locales, the strings <i>uid</i> , <i>gid</i> , <i>euid</i> , <i>egid</i> , and <i>groups</i> may be replaced with more appropriate strings corresponding to the locale.
19437	
19438	
19439	" <i>uid=%u(%s) gid=%u(%s)\n</i> ", <i><real user ID></i> , <i><user-name></i> ,
19440	<i><real group ID></i> , <i><group-name></i>
19441	If the effective and real user IDs do not match, the following shall be inserted immediately before the '\n' character in the previous format:
19442	
19443	" <i>euid=%u(%s)</i> "
19444	with the following arguments added at the end of the argument list:
19445	<i><effective user ID></i> , <i><effective user-name></i>
19446	If the effective and real group IDs do not match, the following shall be inserted directly before the '\n' character in the format string (and after any addition resulting from the effective and real user IDs not matching):
19447	
19448	
19449	" <i>egid=%u(%s)</i> "
19450	with the following arguments added at the end of the argument list:
19451	<i><effective group-ID></i> , <i><effective group name></i>
19452	If the process has supplementary group affiliations or the selected user is allowed to belong to multiple groups, the first shall be added directly before the <newline> in the format string:
19453	
19454	" <i>groups=%u(%s)</i> "
19455	with the following arguments added at the end of the argument list:
19456	<i><supplementary group ID></i> , <i><supplementary group name></i>
19457	and the necessary number of the following added after that for any remaining supplementary group IDs:
19458	
19459	" , <i>%u(%s)</i> "
19460	and the necessary number of the following arguments added at the end of the argument list:
19461	<i><supplementary group ID></i> , <i><supplementary group name></i>

19462 If any of the user ID, group ID, effective user ID, effective group ID, or supplementary/multiple
19463 group IDs cannot be mapped by the system into printable user or group names, the
19464 corresponding "(%s)" and *name* argument shall be omitted from the corresponding format
19465 string.

19466 When any of the options are specified, the output format shall be as described in the OPTIONS
19467 section.

19468 **STDERR**

19469 The standard error shall be used only for diagnostic messages.

19470 **OUTPUT FILES**

19471 None.

19472 **EXTENDED DESCRIPTION**

19473 None.

19474 **EXIT STATUS**

19475 The following exit values shall be returned:

19476 0 Successful completion.

19477 >0 An error occurred.

19478 **CONSEQUENCES OF ERRORS**

19479 Default.

19480 **APPLICATION USAGE**

19481 Output produced by the -G option and by the default case could potentially produce very long
19482 lines on systems that support large numbers of supplementary groups. (On systems with user
19483 and group IDs that are 32-bit integers and with group names with a maximum of 8 bytes per
19484 name, 93 supplementary groups plus distinct effective and real group and user IDs could
19485 theoretically overflow the 2 048-byte {LINE_MAX} text file line limit on the default output case.
19486 It would take about 186 supplementary groups to overflow the 2 048-byte barrier using *id* -G).
19487 This is not expected to be a problem in practice, but in cases where it is a concern, applications
19488 should consider using *fold* -s before postprocessing the output of *id*.

19489 **EXAMPLES**

19490 None.

19491 **RATIONALE**

19492 The functionality provided by the 4 BSD *groups* utility can be simulated using:

19493 `id -Gn [user]`

19494 The 4 BSD command *groups* was considered, but it was not included because it did not provide
19495 the functionality of the *id* utility of the SVID. Also, it was thought that it would be easier to
19496 modify *id* to provide the additional functionality necessary to systems with multiple groups
19497 than to invent another command.

19498 The options -u, -g, -n, and -r were added to ease the use of *id* with shell commands
19499 substitution. Without these options it is necessary to use some preprocessor such as *sed* to select
19500 the desired piece of information. Since output such as that produced by:

19501 `id -u -n`

19502 is frequently wanted, it seemed desirable to add the options.

19503 FUTURE DIRECTIONS

19504 None.

19505 SEE ALSO

19506 *fold*, *logname*, *who*, the System Interfaces volume of IEEE Std 1003.1-2001, *getgid()*, *getgroups()*,
19507 *getuid()*

19508 CHANGE HISTORY

19509 First released in Issue 2.

19510 NAME

19511 ipcrm — remove an XSI message queue, semaphore set, or shared memory segment identifier

19512 SYNOPSIS

19513 XSI ipcrm [-q msgid | -Q msgkey | -s semid | -S semkey |
19514 -m shmid | -M shmkey] ...
19515

19516 DESCRIPTION

19517 The *ipcrm* utility shall remove zero or more message queues, semaphore sets, or shared memory
19518 segments. The interprocess communication facilities to be removed are specified by the options.

19519 Only a user with appropriate privilege shall be allowed to remove an interprocess
19520 communication facility that was not created by or owned by the user invoking *ipcrm*.

19521 OPTIONS

19522 The *ipcrm* facility supports the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
19523 Utility Syntax Guidelines.

19524 The following options shall be supported:

19525 **-q msgid** Remove the message queue identifier *msgid* from the system and destroy the
19526 message queue and data structure associated with it.

19527 **-m shmid** Remove the shared memory identifier *shmid* from the system. The shared memory
19528 segment and data structure associated with it shall be destroyed after the last
19529 detach.

19530 **-s semid** Remove the semaphore identifier *semid* from the system and destroy the set of
19531 semaphores and data structure associated with it.

19532 **-Q msgkey** Remove the message queue identifier, created with key *msgkey*, from the system
19533 and destroy the message queue and data structure associated with it.

19534 **-M shmkey** Remove the shared memory identifier, created with key *shmkey*, from the system.
19535 The shared memory segment and data structure associated with it shall be
19536 destroyed after the last detach.

19537 **-S semkey** Remove the semaphore identifier, created with key *semkey*, from the system and
19538 destroy the set of semaphores and data structure associated with it.

19539 OPERANDS

19540 None.

19541 STDIN

19542 Not used.

19543 INPUT FILES

19544 None.

19545 ENVIRONMENT VARIABLES

19546 The following environment variables shall affect the execution of *ipcrm*:

19547 **LANG** Provide a default value for the internationalization variables that are unset or null.
19548 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19549 Internationalization Variables for the precedence of internationalization variables
19550 used to determine the values of locale categories.)

19551 **LC_ALL** If set to a non-empty string value, override the values of all the other
19552 internationalization variables.

19553	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
19556	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
19559	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
19560	ASYNCHRONOUS EVENTS	
19561		Default.
19562	STDOUT	
19563		Not used.
19564	STDERR	
19565		The standard error shall be used only for diagnostic messages.
19566	OUTPUT FILES	
19567		None.
19568	EXTENDED DESCRIPTION	
19569		None.
19570	EXIT STATUS	
19571		The following exit values shall be returned:
19572	0	Successful completion.
19573	>0	An error occurred.
19574	CONSEQUENCES OF ERRORS	
19575		Default.
19576	APPLICATION USAGE	
19577		None.
19578	EXAMPLES	
19579		None.
19580	RATIONALE	
19581		None.
19582	FUTURE DIRECTIONS	
19583		None.
19584	SEE ALSO	
19585		<i>ipcs</i> , the System Interfaces volume of IEEE Std 1003.1-2001, <i>msgctl()</i> , <i>semctl()</i> , <i>shmctl()</i>
19586	CHANGE HISTORY	
19587		First released in Issue 5.

19588 NAME

19589 ipcs — report XSI interprocess communication facilities status

19590 SYNOPSIS

19591 XSI ipcs [-qms][-a | -bcopt]

19592

19593 DESCRIPTION

19594 The *ipcs* utility shall write information about active interprocess communication facilities.

19595 Without options, information shall be written in short format for message queues, shared
19596 memory segments, and semaphore sets that are currently active in the system. Otherwise, the
19597 information that is displayed is controlled by the options specified.

19598 OPTIONS

19599 The *ipcs* facility supports the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
19600 Utility Syntax Guidelines.

19601 The *ipcs* utility accepts the following options:

19602 **-q** Write information about active message queues.

19603 **-m** Write information about active shared memory segments.

19604 **-s** Write information about active semaphore sets.

19605 If **-q**, **-m**, or **-s** are specified, only information about those facilities shall be written. If none of
19606 these three are specified, information about all three shall be written subject to the following
19607 options:

19608 **-a** Use all print options. (This is a shorthand notation for **-b**, **-c**, **-o**, **-p**, and **-t**.)

19609 **-b** Write information on maximum allowable size. (Maximum number of bytes in
19610 messages on queue for message queues, size of segments for shared memory, and
19611 number of semaphores in each set for semaphores.)

19612 **-c** Write creator's user name and group name; see below.

19613 **-o** Write information on outstanding usage. (Number of messages on queue and total
19614 number of bytes in messages on queue for message queues, and number of
19615 processes attached to shared memory segments.)

19616 **-p** Write process number information. (Process ID of the last process to send a
19617 message and process ID of the last process to receive a message on message
19618 queues, process ID of the creating process, and process ID of the last process to
19619 attach or detach on shared memory segments.)

19620 **-t** Write time information. (Time of the last control operation that changed the access
19621 permissions for all facilities, time of the last *msgsnd()* and *msgrcv()* operations on
19622 message queues, time of the last *shmat()* and *shmdt()* operations on shared
19623 memory, and time of the last *semop()* operation on semaphores.)

19624 OPERANDS

19625 None.

19626 STDIN

19627 Not used.

19628 INPUT FILES

- 19629 • The group database
 19630 • The user database

19631 ENVIRONMENT VARIABLES

19632 The following environment variables shall affect the execution of *ipcs*:

- 19633 *LANG* Provide a default value for the internationalization variables that are unset or null.
 19634 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
 19635 Internationalization Variables for the precedence of internationalization variables
 19636 used to determine the values of locale categories.)
- 19637 *LC_ALL* If set to a non-empty string value, override the values of all the other
 19638 internationalization variables.
- 19639 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
 19640 characters (for example, single-byte as opposed to multi-byte characters in
 19641 arguments).
- 19642 *LC_MESSAGES*
 19643 Determine the locale that should be used to affect the format and contents of
 19644 diagnostic messages written to standard error.
- 19645 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.
- 19646 *TZ* Determine the timezone for the date and time strings written by *ipcs*. If *TZ* is unset
 19647 or null, an unspecified default timezone shall be used.

19648 ASYNCHRONOUS EVENTS

19649 Default.

19650 STDOUT

19651 An introductory line shall be written with the format:

19652 "IPC status from %s as of %s\n", <source>, <date>

19653 where <source> indicates the source used to gather the statistics and <date> is the information
 19654 that would be produced by the *date* command when invoked in the POSIX locale.

19655 The *ipcs* utility then shall create up to three reports depending upon the **-q**, **-m**, and **-s** options.
 19656 The first report shall indicate the status of message queues, the second report shall indicate the
 19657 status of shared memory segments, and the third report shall indicate the status of semaphore
 19658 sets.

19659 If the corresponding facility is not installed or has not been used since the last reboot, then the
 19660 report shall be written out in the format:

19661 "%s facility not in system.\n", <facility>

19662 where <facility> is *Message Queue*, *Shared Memory*, or *Semaphore*, as appropriate. If the facility has
 19663 been installed and has been used since the last reboot, column headings separated by one or
 19664 more spaces and followed by a <newline> shall be written as indicated below followed by the
 19665 facility name written out using the format:

19666 ">%s:\n", <facility>

19667 where <facility> is *Message Queues*, *Shared Memory*, or *Semaphores*, as appropriate. On the second
 19668 and third reports the column headings need not be written if the last column headings written
 19669 already provide column headings for all information in that report.

19670 The column headings provided in the first column below and the meaning of the information in
19671 those columns shall be given in order below; the letters in parentheses indicate the options that
19672 shall cause the corresponding column to appear; "all" means that the column shall always
19673 appear. Each column is separated by one or more <space>s. Note that these options only
19674 determine what information is provided for each report; they do not determine which reports
19675 are written.

19676	T	(all)	Type of facility:
19677		q	Message queue.
19678		m	Shared memory segment.
19679		s	Semaphore.
19680			This field is a single character written using the format %c.
19681	ID	(all)	The identifier for the facility entry. This field shall be written using the format 19682 %d.
19683	KEY	(all)	The key used as an argument to <i>msgget()</i> , <i>semget()</i> , or <i>shmget()</i> to create the 19684 facility entry. Note: The key of a shared memory segment is changed to IPC_PRIVATE when 19685 the segment has been removed until all processes attached to the segment 19686 detach it.
19688			This field shall be written using the format 0x% ₂ x.
19689	MODE	(all)	The facility access modes and flags. The mode shall consist of 11 characters 19690 that are interpreted as follows. The first character shall be: 19692 S If a process is waiting on a <i>msgsnd()</i> operation. 19693 – If the above is not true. The second character shall be: 19695 R If a process is waiting on a <i>msgrecv()</i> operation. 19696 C or – If the associated shared memory segment is to be cleared when the 19697 first attach operation is executed. 19698 – If none of the above is true. 19699 The next nine characters shall be interpreted as three sets of three bits each. 19700 The first set refers to the owner's permissions; the next to permissions of 19701 others in the usergroup of the facility entry; and the last to all others. Within 19702 each set, the first character indicates permission to read, the second character 19703 indicates permission to write or alter the facility entry, and the last character is 19704 a minus sign ('-'). 19705 The permissions shall be indicated as follows: 19706 r If read permission is granted. 19707 w If write permission is granted. 19708 a If alter permission is granted. 19709 – If the indicated permission is not granted.

19710	The first character following the permissions specifies if there is an alternate or additional access control method associated with the facility. If there is no alternate or additional access control method associated with the facility, a single <space> shall be written; otherwise, another printable character is written.	
19715	OWNER (all) The user name of the owner of the facility entry. If the user name of the owner is found in the user database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the user ID of the owner shall be written using the format %d.	
19719	GROUP (all) The group name of the owner of the facility entry. If the group name of the owner is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the owner shall be written using the format %d.	
19723	The following nine columns shall be only written out for message queues:	
19724	CREATOR (a,c) The user name of the creator of the facility entry. If the user name of the creator is found in the user database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the user ID of the creator shall be written using the format %d.	
19728	CGROUP (a,c) The group name of the creator of the facility entry. If the group name of the creator is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the creator shall be written using the format %d.	
19732	CBYTES (a,o) The number of bytes in messages currently outstanding on the associated message queue. This field shall be written using the format %d.	
19734	QNUM (a,o) The number of messages currently outstanding on the associated message queue. This field shall be written using the format %d.	
19736	QBYTES (a,b) The maximum number of bytes allowed in messages outstanding on the associated message queue. This field shall be written using the format %d.	
19738	LSPID (a,p) The process ID of the last process to send a message to the associated queue. This field shall be written using the format: " <i>%d</i> ", < <i>pid</i> >	
19741	where < <i>pid</i> > is 0 if no message has been sent to the corresponding message queue; otherwise, < <i>pid</i> > shall be the process ID of the last process to send a message to the queue.	
19744	LRPID (a,p) The process ID of the last process to receive a message from the associated queue. This field shall be written using the format: " <i>%d</i> ", < <i>pid</i> >	
19747	where < <i>pid</i> > is 0 if no message has been received from the corresponding message queue; otherwise, < <i>pid</i> > shall be the process ID of the last process to receive a message from the queue.	
19750	STIME (a,t) The time the last message was sent to the associated queue. If a message has been sent to the corresponding message queue, the hour, minute, and second of the last time a message was sent to the queue shall be written using the format %d.%2.2d.%2.2d. Otherwise, the format "no-entry" shall be written.	

19755	RTIME	(a,t)	The time the last message was received from the associated queue. If a message has been received from the corresponding message queue, the hour, minute, and second of the last time a message was received from the queue shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.
19760			The following eight columns shall be only written out for shared memory segments.
19761	CREATOR	(a,c)	The user of the creator of the facility entry. If the user name of the creator is found in the user database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the user ID of the creator shall be written using the format %d.
19765	CGROUP	(a,c)	The group name of the creator of the facility entry. If the group name of the creator is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the creator shall be written using the format %d.
19769	NATTCH	(a,o)	The number of processes attached to the associated shared memory segment. This field shall be written using the format %d.
19771	SEGSZ	(a,b)	The size of the associated shared memory segment. This field shall be written using the format %d.
19773	CPID	(a,p)	The process ID of the creator of the shared memory entry. This field shall be written using the format %d.
19775	LPID	(a,p)	The process ID of the last process to attach or detach the shared memory segment. This field shall be written using the format: "%d", <pid> where <pid> is 0 if no process has attached the corresponding shared memory segment; otherwise, <pid> shall be the process ID of the last process to attach or detach the segment.
19781	ATIME	(a,t)	The time the last attach on the associated shared memory segment was completed. If the corresponding shared memory segment has ever been attached, the hour, minute, and second of the last time the segment was attached shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.
19786	DTIME	(a,t)	The time the last detach on the associated shared memory segment was completed. If the corresponding shared memory segment has ever been detached, the hour, minute, and second of the last time the segment was detached shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.
19791			The following four columns shall be only written out for semaphore sets:
19792	CREATOR	(a,c)	The user of the creator of the facility entry. If the user name of the creator is found in the user database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the user ID of the creator shall be written using the format %d.
19796	CGROUP	(a,c)	The group name of the creator of the facility entry. If the group name of the creator is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the creator shall be written using the format %d.

19800	NSEMS	(a,b)	The number of semaphores in the set associated with the semaphore entry.
19801			This field shall be written using the format %d.
19802	OTIME	(a,t)	The time the last semaphore operation on the set associated with the semaphore entry was completed. If a semaphore operation has ever been performed on the corresponding semaphore set, the hour, minute, and second of the last semaphore operation on the semaphore set shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.
19803			
19804			
19805			
19806			
19807			
19808			The following column shall be written for all three reports when it is requested:
19809	CTIME	(a,t)	The time the associated entry was created or changed. The hour, minute, and second of the time when the associated entry was created shall be written using the format %d:%2.2d:%2.2d.
19810			
19811			
19812	STDERR		
19813			The standard error shall be used only for diagnostic messages.
19814	OUTPUT FILES		
19815			None.
19816	EXTENDED DESCRIPTION		
19817			None.
19818	EXIT STATUS		
19819			The following exit values shall be returned:
19820		0	Successful completion.
19821		>0	An error occurred.
19822	CONSEQUENCES OF ERRORS		
19823			Default.
19824	APPLICATION USAGE		
19825			Things can change while <i>ipcs</i> is running; the information it gives is guaranteed to be accurate only when it was retrieved.
19826			
19827	EXAMPLES		
19828			None.
19829	RATIONALE		
19830			None.
19831	FUTURE DIRECTIONS		
19832			None.
19833	SEE ALSO		
19834			The System Interfaces volume of IEEE Std 1003.1-2001, <i>msgrecv()</i> , <i>msgsnd()</i> , <i>semget()</i> , <i>semop()</i> ,
19835			<i>shmat()</i> , <i>shmdt()</i> , <i>shmget()</i>
19836	CHANGE HISTORY		
19837			First released in Issue 5.
19838	Issue 6		
19839			The Open Group Corrigendum U020/1 is applied, correcting the SYNOPSIS.
19840			The Open Group Corrigenda U032/1 and U032/2 are applied, clarifying the output format.
19841			The Open Group Base Resolution bwg98-004 is applied.

19842 NAME

19843 jobs — display status of jobs in the current session

19844 SYNOPSIS

19845 UP jobs [-l | -p][job_id...]

19846

19847 DESCRIPTION

19848 The *jobs* utility shall display the status of jobs that were started in the current shell environment;
19849 see Section 2.12 (on page 61).

19850 When *jobs* reports the termination status of a job, the shell shall remove its process ID from the
19851 list of those “known in the current shell execution environment”; see Section 2.9.3.1 (on page
19852 50).

19853 OPTIONS

19854 The *jobs* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
19855 12.2, Utility Syntax Guidelines.

19856 The following options shall be supported:

19857 **-l** (The letter ell.) Provide more information about each job listed. This information
19858 shall include the job number, current job, process group ID, state, and the
19859 command that formed the job.

19860 **-p** Display only the process IDs for the process group leaders of the selected jobs.

19861 By default, the *jobs* utility shall display the status of all stopped jobs, running background jobs
19862 and all jobs whose status has changed and have not been reported by the shell.

19863 OPERANDS

19864 The following operand shall be supported:

19865 *job_id* Specifies the jobs for which the status is to be displayed. If no *job_id* is given, the
19866 status information for all jobs shall be displayed. The format of *job_id* is described
19867 in the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.203, Job Control
19868 Job ID.

19869 STDIN

19870 Not used.

19871 INPUT FILES

19872 None.

19873 ENVIRONMENT VARIABLES

19874 The following environment variables shall affect the execution of *jobs*:

19875 **LANG** Provide a default value for the internationalization variables that are unset or null.
19876 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
19877 Internationalization Variables for the precedence of internationalization variables
19878 used to determine the values of locale categories.)

19879 **LC_ALL** If set to a non-empty string value, override the values of all the other
19880 internationalization variables.

19881 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
19882 characters (for example, single-byte as opposed to multi-byte characters in
19883 arguments).

19884 **LC_MESSAGES**

19885 Determine the locale that should be used to affect the format and contents of

19886 diagnostic messages written to standard error and informative messages written to
 19887 standard output.

19888 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

19889 **ASYNCHRONOUS EVENTS**

19890 Default.

19891 **STDOUT**

19892 If the **-p** option is specified, the output shall consist of one line for each process ID:

19893 "*%d\n*", <process ID>

19894 Otherwise, if the **-l** option is not specified, the output shall be a series of lines of the form:

19895 "*[%d] %c %s %s\n*", <job-number>, <current>, <state>, <command>

19896 where the fields shall be as follows:

19897 <current> The character '+' identifies the job that would be used as a default for the *fg* or *bg*
 19898 utilities; this job can also be specified using the *job_id* %+ or "%". The character
 19899 '-' identifies the job that would become the default if the current default job were
 19900 to exit; this job can also be specified using the *job_id* %- . For other jobs, this field is
 19901 a <space>. At most one job can be identified with '+' and at most one job can be
 19902 identified with '-'. If there is any suspended job, then the current job shall be a
 19903 suspended job. If there are at least two suspended jobs, then the previous job also
 19904 shall be a suspended job.

19905 <job-number> A number that can be used to identify the process group to the *wait*, *fg*, *bg*, and *kill*
 19906 utilities. Using these utilities, the job can be identified by prefixing the job number
 19907 with '%'.

19908 <state> One of the following strings (in the POSIX locale):

19909 **Running** Indicates that the job has not been suspended by a signal and has not
 19910 exited.

19911 **Done** Indicates that the job completed and returned exit status zero.

19912 **Done(code)** Indicates that the job completed normally and that it exited with the
 19913 specified non-zero exit status, *code*, expressed as a decimal number.

19914 **Stopped** Indicates that the job was suspended by the SIGTSTP signal.

19915 **Stopped (SIGTSTP)**
 19916 Indicates that the job was suspended by the SIGTSTP signal.

19917 **Stopped (SIGSTOP)**
 19918 Indicates that the job was suspended by the SIGSTOP signal.

19919 **Stopped (SIGTTIN)**
 19920 Indicates that the job was suspended by the SIGTTIN signal.

19921 **Stopped (SIGTTOU)**
 19922 Indicates that the job was suspended by the SIGTTOU signal.

19923 The implementation may substitute the string **Suspended** in place of **Stopped**. If
 19924 the job was terminated by a signal, the format of <state> is unspecified, but it shall
 19925 be visibly distinct from all of the other <state> formats shown here and shall
 19926 indicate the name or description of the signal causing the termination.

- 19927 *<command>* The associated command that was given to the shell.
- 19928 If the **-l** option is specified, a field containing the process group ID shall be inserted before the
19929 *<state>* field. Also, more processes in a process group may be output on separate lines, using
19930 only the process ID and *<command>* fields.
- 19931 **STDERR**
- 19932 The standard error shall be used only for diagnostic messages.
- 19933 **OUTPUT FILES**
- 19934 None.
- 19935 **EXTENDED DESCRIPTION**
- 19936 None.
- 19937 **EXIT STATUS**
- 19938 The following exit values shall be returned:
- 19939 0 Successful completion.
- 19940 >0 An error occurred.
- 19941 **CONSEQUENCES OF ERRORS**
- 19942 Default.
- 19943 **APPLICATION USAGE**
- 19944 The **-p** option is the only portable way to find out the process group of a job because different
19945 implementations have different strategies for defining the process group of the job. Usage such
19946 as *\$jobs -p* provides a way of referring to the process group of the job in an implementation-
19947 independent way.
- 19948 The *jobs* utility does not work as expected when it is operating in its own utility execution
19949 environment because that environment has no applicable jobs to manipulate. See the
19950 APPLICATION USAGE section for *bg*. For this reason, *jobs* is generally implemented as a shell
19951 regular built-in.
- 19952 **EXAMPLES**
- 19953 None.
- 19954 **RATIONALE**
- 19955 Both "`%%`" and "`%+`" are used to refer to the current job. Both forms are of equal validity—the
19956 "`%%`" mirroring "`$$`" and "`%+`" mirroring the output of *jobs*. Both forms reflect historical
19957 practice of the KornShell and the C shell with job control.
- 19958 The job control features provided by *bg*, *fg*, and *jobs* are based on the KornShell. The standard
19959 developers examined the characteristics of the C shell versions of these utilities and found that
19960 differences exist. Despite widespread use of the C shell, the KornShell versions were selected for
19961 this volume of IEEE Std 1003.1-2001 to maintain a degree of uniformity with the rest of the
19962 KornShell features selected (such as the very popular command line editing features).
- 19963 The *jobs* utility is not dependent on the job control option, as are the seemingly related *bg* and *fg*
19964 utilities because *jobs* is useful for examining background jobs, regardless of the condition of job
19965 control. When the user has invoked a *set +m* command and job control has been turned off, *jobs*
19966 can still be used to examine the background jobs associated with that current session. Similarly,
19967 *kill* can then be used to kill background jobs with *kill% <background job number>*.
- 19968 The output for terminated jobs is left unspecified to accommodate various historical systems.
19969 The following formats have been witnessed:

- 19970 1. **Killed**(*signal name*)
19971 2. *signal name*
19972 3. *signal name(coredump)*
19973 4. *signal description– core dumped*
- 19974 Most users should be able to understand these formats, although it means that applications have
19975 trouble parsing them.
- 19976 The calculation of job IDs was not described since this would suggest an implementation, which
19977 may impose unnecessary restrictions.
- 19978 In an early proposal, a **-n** option was included to “Display the status of jobs that have changed,
19979 exited, or stopped since the last status report”. It was removed because the shell always writes
19980 any changed status of jobs before each prompt.
- 19981 **FUTURE DIRECTIONS**
19982 None.
- 19983 **SEE ALSO**
19984 Section 2.12 (on page 61), *bg*, *fg*, *kill*, *wait*
- 19985 **CHANGE HISTORY**
19986 First released in Issue 4.
- 19987 **Issue 6**
19988 This utility is marked as part of the User Portability Utilities option.
19989 The JC shading is removed as job control is mandatory in this issue.

19990 NAME

19991 join — relational database operator

19992 SYNOPSIS

19993 join [-a *file_number* | -v *file_number*] [-e *string*] [-o *list*] [-t *char*]
19994 [-1 *field*] [-2 *field*] *file1* *file2*

19995 DESCRIPTION

19996 The *join* utility shall perform an equality join on the files *file1* and *file2*. The joined files shall be
19997 written to the standard output.

19998 The join field is a field in each file on which the files are compared. The *join* utility shall write
19999 one line in the output for each pair of lines in *file1* and *file2* that have identical join fields. The
20000 output line by default shall consist of the join field, then the remaining fields from *file1*, then the
20001 remaining fields from *file2*. This format can be changed by using the **-o** option (see below). The
20002 **-a** option can be used to add unmatched lines to the output. The **-v** option can be used to output
20003 only unmatched lines.

20004 The files *file1* and *file2* shall be ordered in the collating sequence of *sort -b* on the fields on which
20005 they shall be joined, by default the first in each line. All selected output shall be written in the
20006 same collating sequence.

20007 The default input field separators shall be <blank>s. In this case, multiple separators shall count
20008 as one field separator, and leading separators shall be ignored. The default output field separator
20009 shall be a <space>.

20010 The field separator and collating sequence can be changed by using the **-t** option (see below).

20011 If the same key appears more than once in either file, all combinations of the set of remaining
20012 fields in *file1* and the set of remaining fields in *file2* are output in the order of the lines
20013 encountered.

20014 If the input files are not in the appropriate collating sequence, the results are unspecified.

20015 OPTIONS

20016 The *join* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
20017 12.2, Utility Syntax Guidelines.

20018 The following options shall be supported:

20019 **-a** *file_number*

20020 Produce a line for each unpairable line in file *file_number*, where *file_number* is 1 or
20021 2, in addition to the default output. If both **-a1** and **-a2** are specified, all unpairable
20022 lines shall be output.

20023 **-e** *string*

Replace empty output fields in the list selected by **-o** with the string *string*.

20024 **-o** *list*

20025 Construct the output line to comprise the fields specified in *list*, each element of
which shall have one of the following two forms:

- 20026 1. *file_number.field*, where *file_number* is a file number and *field* is a decimal
20027 integer field number

- 20028 2. 0 (zero), representing the join field

20029 The elements of *list* shall be either comma-separated or <blank>-separated, as
20030 specified in Guideline 8 of the Base Definitions volume of IEEE Std 1003.1-2001,
20031 Section 12.2, Utility Syntax Guidelines. The fields specified by *list* shall be written
20032 for all selected output lines. Fields selected by *list* that do not appear in the input
20033 shall be treated as empty output fields. (See the **-e** option.) Only specifically

20034 requested fields shall be written. The application shall ensure that *list* is a single
20035 command line argument.

20036 **-t char** Use character *char* as a separator, for both input and output. Every appearance of
20037 *char* in a line shall be significant. When this option is specified, the collating
20038 sequence shall be the same as *sort* without the **-b** option.

20039 **-v file_number** Instead of the default output, produce a line only for each unpairable line in
20040 *file_number*, where *file_number* is 1 or 2. If both **-v1** and **-v2** are specified, all
20041 unpairable lines shall be output.

20043 **-1 field** Join on the *field*th field of file 1. Fields are decimal integers starting with 1.

20044 **-2 field** Join on the *field*th field of file 2. Fields are decimal integers starting with 1.

20045 OPERANDS

20046 The following operands shall be supported:

20047 *file1, file2* A pathname of a file to be joined. If either of the *file1* or *file2* operands is '**-**', the
20048 standard input shall be used in its place.

20049 STDIN

20050 The standard input shall be used only if the *file1* or *file2* operand is '**-**'. See the INPUT FILES
20051 section.

20052 INPUT FILES

20053 The input files shall be text files.

20054 ENVIRONMENT VARIABLES

20055 The following environment variables shall affect the execution of *join*:

20056 **LANG** Provide a default value for the internationalization variables that are unset or null.
20057 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
20058 Internationalization Variables for the precedence of internationalization variables
20059 used to determine the values of locale categories.)

20060 **LC_ALL** If set to a non-empty string value, override the values of all the other
20061 internationalization variables.

20062 **LC_COLLATE** Determine the locale of the collating sequence *join* expects to have been used when
20063 the input files were sorted.

20065 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
20066 characters (for example, single-byte as opposed to multi-byte characters in
20067 arguments and input files).

20068 **LC_MESSAGES** Determine the locale that should be used to affect the format and contents of
20069 diagnostic messages written to standard error.

20071 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

20072 ASYNCHRONOUS EVENTS

20073 Default.

20074 **STDOUT**

20075 The *join* utility output shall be a concatenation of selected character fields. When the **-o** option
20076 is not specified, the output shall be:

20077 "%s%s\n", <join field>, <other file1 fields>,
20078 <other file2 fields>

20079 If the join field is not the first field in a file, the <other file fields> for that file shall be:

20080 <fields preceding join field>, <fields following join field>

20081 When the **-o** option is specified, the output format shall be:

20082 ">%s\n", <concatenation of fields>

20083 where the concatenation of fields is described by the **-o** option, above.

20084 For either format, each field (except the last) shall be written with its trailing separator character.
20085 If the separator is the default (<blank>s), a single <space> shall be written after each field
20086 (except the last).

20087 **STDERR**

20088 The standard error shall be used only for diagnostic messages.

20089 **OUTPUT FILES**

20090 None.

20091 **EXTENDED DESCRIPTION**

20092 None.

20093 **EXIT STATUS**

20094 The following exit values shall be returned:

20095 0 All input files were output successfully.

20096 >0 An error occurred.

20097 **CONSEQUENCES OF ERRORS**

20098 Default.

20099 **APPLICATION USAGE**

20100 Pathnames consisting of numeric digits or of the form *string.string* should not be specified
20101 directly following the **-o** list.

20102 **EXAMPLES**

20103 The **-o 0** field essentially selects the union of the join fields. For example, given file **phone**:

20104 !Name Phone Number
20105 Don +1 123-456-7890
20106 Hal +1 234-567-8901
20107 Yasushi +2 345-678-9012

20108 and file **fax**:

20109 !Name Fax Number
20110 Don +1 123-456-7899
20111 Keith +1 456-789-0122
20112 Yasushi +2 345-678-9011

20113 (where the large expanses of white space are meant to each represent a single <tab>), the
20114 command:

20115 join -t "<tab>" -a 1 -a 2 -e '(unknown)' -o 0,1.2,2.2 phone fax
 20116 would produce:

20117 !Name	20118 Phone Number	20119 Fax Number
Don	+1 123-456-7890	+1 123-456-7899
Hal	+1 234-567-8901	(unknown)
Keith	(unknown)	+1 456-789-0122
Yasushi	+2 345-678-9012	+2 345-678-9011

20122 Multiple instances of the same key will produce combinatorial results. The following:

20123 fa:
 20124 a x
 20125 a y
 20126 a z
 20127 fb:
 20128 a p

20129 will produce:

20130 a x p
 20131 a y p
 20132 a z p

20133 And the following:

20134 fa:
 20135 a b c
 20136 a d e
 20137 fb:
 20138 a w x
 20139 a y z
 20140 a o p

20141 will produce:

20142 a b c w x
 20143 a b c y z
 20144 a b c o p
 20145 a d e w x
 20146 a d e y z
 20147 a d e o p

20148 RATIONALE

20149 The -e option is only effective when used with -o because, unless specific fields are identified
 20150 using -o, join is not aware of what fields might be empty. The exception to this is the join field,
 20151 but identifying an empty join field with the -e string is not historical practice and some scripts
 20152 might break if this were changed.

20153 The 0 field in the -o list was adopted from the Tenth Edition version of join to satisfy
 20154 international objections that the join in the base documents does not support the “full join” or
 20155 “outer join” described in relational database literature. Although it has been possible to include
 20156 a join field in the output (by default, or by field number using -o), the join field could not be
 20157 included for an unpaired line selected by -a. The -o 0 field essentially selects the union of the
 20158 join fields.

20159 This sort of outer join was not possible with the join commands in the base documents. The -o 0
 20160 field was chosen because it is an upwards-compatible change for applications. An alternative

20161 was considered: have the join field represent the union of the fields in the files (where they are
20162 identical for matched lines, and one or both are null for unmatched lines). This was not adopted
20163 because it would break some historical applications.

20164 The ability to specify *file2* as – is not historical practice; it was added for completeness.

20165 The –v option is not historical practice, but was considered necessary because it permitted the
20166 writing of *only* those lines that do not match on the join field, as opposed to the –a option, which
20167 prints both lines that do and do not match. This additional facility is parallel with the –v option
20168 of *grep*.

20169 Some historical implementations have been encountered where a blank line in one of the input
20170 files was considered to be the end of the file; the description in this volume of
20171 IEEE Std 1003.1-2001 does not cite this as an allowable case.

20172 **FUTURE DIRECTIONS**

20173 None.

20174 **SEE ALSO**

20175 *awk, comm, sort, uniq*

20176 **CHANGE HISTORY**

20177 First released in Issue 2.

20178 **Issue 6**

20179 The obsolescent –j options and the multi-argument –o option are withdrawn in this issue.

20180 The normative text is reworded to avoid use of the term “must” for application requirements.

20181 NAME

20182 kill — terminate or signal processes

20183 SYNOPSIS

20184 kill -s *signal_name* *pid* ...20185 kill -l [*exit_status*]20186 XSI kill [-*signal_name*] *pid* ...20187 kill [-*signal_number*] *pid* ...

20188

20189 DESCRIPTION

20190 The *kill* utility shall send a signal to the process or processes specified by each *pid* operand.20191 For each *pid* operand, the *kill* utility shall perform actions equivalent to the *kill()* function
20192 defined in the System Interfaces volume of IEEE Std 1003.1-2001 called with the following
20193 arguments:

- 20194 • The value of the *pid* operand shall be used as the *pid* argument.
- 20195 • The *sig* argument is the value specified by the -s option, -*signal_number* option, or the
20196 -*signal_name* option, or by SIGTERM, if none of these options is specified.

20197 OPTIONS

20198 The *kill* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
20199 XSI 12.2, Utility Syntax Guidelines, except that in the last two SYNOPSIS forms, the -*signal_number*
20200 and -*signal_name* options are usually more than a single character.

20201 The following options shall be supported:

20202 -l (The letter ell.) Write all values of *signal_name* supported by the implementation, if
20203 no operand is given. If an *exit_status* operand is given and it is a value of the '?'
20204 shell special parameter (see Section 2.5.2 (on page 34) and *wait*) corresponding to a
20205 process that was terminated by a signal, the *signal_name* corresponding to the
20206 signal that terminated the process shall be written. If an *exit_status* operand is
20207 given and it is the unsigned decimal integer value of a signal number, the
20208 *signal_name* (the symbolic constant name without the SIG prefix defined in the
20209 Base Definitions volume of IEEE Std 1003.1-2001) corresponding to that signal
20210 shall be written. Otherwise, the results are unspecified.

20211 -s *signal_name*

20212 Specify the signal to send, using one of the symbolic names defined in the
20213 <signal.h> header. Values of *signal_name* shall be recognized in a case-independent
20214 fashion, without the SIG prefix. In addition, the symbolic name 0 shall be
20215 recognized, representing the signal value zero. The corresponding signal shall be
20216 sent instead of SIGTERM.

20217 XSI -*signal_name*

20218 Equivalent to -s *signal_name*.

20219 XSI -*signal_number*

20220 Specify a non-negative decimal integer, *signal_number*, representing the signal to
20221 be used instead of SIGTERM, as the *sig* argument in the effective call to *kill()*. The
20222 correspondence between integer values and the *sig* value used is shown in the
20223 following table.

20224 The effects of specifying any *signal_number* other than those listed in the table are
20225 undefined.

20226

20227

	<i>signal_number</i>	<i>sig Value</i>
20228 XSI	0	0
20229	1	SIGHUP
20230	2	SIGINT
20231	3	SIGQUIT
20232	6	SIGABRT
20233	9	SIGKILL
20234	14	SIGALRM
20235	15	SIGTERM

20236

20237

If the first argument is a negative integer, it shall be interpreted as a *-signal_number* option, not as a negative *pid* operand specifying a process group.

20238 **OPERANDS**

20239

The following operands shall be supported:

20240

pid One of the following:

20241

20242

20243

20244

20245

20246

20247

1. A decimal integer specifying a process or process group to be signaled. The process or processes selected by positive, negative, and zero values of the *pid* operand shall be as described for the *kill()* function. If process number 0 is specified, all processes in the current process group shall be signaled. For the effects of negative *pid* numbers, see the *kill()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001. If the first *pid* operand is negative, it should be preceded by "—" to keep it from being interpreted as an option.
2. A job control job ID (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.203, Job Control Job ID) that identifies a background process group to be signaled. The job control job ID notation is applicable only for invocations of *kill* in the current shell execution environment; see Section 2.12 (on page 61).

20253

exit_status A decimal integer specifying a signal number or the exit status of a process terminated by a signal.

20255 **STDIN**

20256

Not used.

20257 **INPUT FILES**

20258

None.

20259 **ENVIRONMENT VARIABLES**

20260

The following environment variables shall affect the execution of *kill*:

20261

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

20265

20266

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

20267

20268

20269

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

20270

LC_MESSAGES

20271

Determine the locale that should be used to affect the format and contents of

20272 diagnostic messages written to standard error.

20273 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

20274 **ASYNCHRONOUS EVENTS**

20275 Default.

20276 **STDOUT**

20277 When the **-l** option is not specified, the standard output shall not be used.

20278 When the **-l** option is specified, the symbolic name of each signal shall be written in the
20279 following format:

20280 "%s%c", <signal_name>, <separator>

20281 where the <signal_name> is in uppercase, without the **SIG** prefix, and the <separator> shall be
20282 either a <newline> or a <space>. For the last signal written, <separator> shall be a <newline>.

20283 When both the **-l** option and *exit_status* operand are specified, the symbolic name of the
20284 corresponding signal shall be written in the following format:

20285 "%s\n", <signal_name>

20286 **STDERR**

20287 The standard error shall be used only for diagnostic messages.

20288 **OUTPUT FILES**

20289 None.

20290 **EXTENDED DESCRIPTION**

20291 None.

20292 **EXIT STATUS**

20293 The following exit values shall be returned:

20294 0 At least one matching process was found for each *pid* operand, and the specified signal was
20295 successfully processed for at least one matching process.

20296 >0 An error occurred.

20297 **CONSEQUENCES OF ERRORS**

20298 Default.

20299 **APPLICATION USAGE**

20300 Process numbers can be found by using *ps*.

20301 The job control job ID notation is not required to work as expected when *kill* is operating in its
20302 own utility execution environment. In either of the following examples:

20303 nohup kill %1 &
20304 system("kill %1");

20305 the *kill* operates in a different environment and does not share the shell's understanding of job
20306 numbers.

20307 **EXAMPLES**

20308 Any of the commands:

20309 kill -9 100 -165

20310 kill -s kill 100 -165

20311 kill -s KILL 100 -165

20312 sends the SIGKILL signal to the process whose process ID is 100 and to all processes whose
20313 process group ID is 165, assuming the sending process has permission to send that signal to the
20314 specified processes, and that they exist.

20315 The System Interfaces volume of IEEE Std 1003.1-2001 and this volume of IEEE Std 1003.1-2001
20316 do not require specific signal numbers for any *signal_names*. Even the **-signal_number** option
20317 provides symbolic (although numeric) names for signals. If a process is terminated by a signal,
20318 its exit status indicates the signal that killed it, but the exact values are not specified. The **kill -l**
20319 option, however, can be used to map decimal signal numbers and exit status values into the
20320 name of a signal. The following example reports the status of a terminated job:

```
20321 job
20322 stat=$?
20323 if [ $stat -eq 0 ]
20324 then
20325     echo job completed successfully.
20326 elif [ $stat -gt 128 ]
20327 then
20328     echo job terminated by signal SIG$(kill -l $stat).
20329 else
20330     echo job terminated with error code $stat.
20331 fi
```

20332 To send the default signal to a process group (say 123), an application should use a command
20333 similar to one of the following:

```
20334 kill -TERM -123
20335 kill -- -123
```

20336 RATIONALE

20337 The **-l** option originated from the C shell, and is also implemented in the KornShell. The C shell
20338 output can consist of multiple output lines because the signal names do not always fit on a
20339 single line on some terminal screens. The KornShell output also included the implementation-
20340 defined signal numbers and was considered by the standard developers to be too difficult for
20341 scripts to parse conveniently. The specified output format is intended not only to accommodate
20342 the historical C shell output, but also to permit an entirely vertical or entirely horizontal listing
20343 on systems for which this is appropriate.

20344 An early proposal invented the name SIGNULL as a *signal_name* for signal 0 (used by the System
20345 Interfaces volume of IEEE Std 1003.1-2001 to test for the existence of a process without sending it
20346 a signal). Since the *signal_name* 0 can be used in this case unambiguously, SIGNULL has been
20347 removed.

20348 An early proposal also required symbolic *signal_names* to be recognized with or without the **SIG**
20349 prefix. Historical versions of **kill** have not written the **SIG** prefix for the **-l** option and have not
20350 recognized the **SIG** prefix on *signal_names*. Since neither applications portability nor ease-of-use
20351 would be improved by requiring this extension, it is no longer required.

20352 To avoid an ambiguity of an initial negative number argument specifying either a signal number
20353 or a process group, IEEE Std 1003.1-2001 mandates that it is always considered the former by
20354 implementations that support the XSI option. It also requires that conforming applications
20355 always use the **--** options terminator argument when specifying a process group, unless an
20356 option is also specified.

20357 The **-s** option was added in response to international interest in providing some form of **kill** that
20358 meets the Utility Syntax Guidelines.

20359 The job control job ID notation is not required to work as expected when *kill* is operating in its
20360 own utility execution environment. In either of the following examples:

20361 nohup kill %1 &
20362 system("kill %1");
20363 the *kill* operates in a different environment and does not understand how the shell has managed
20364 its job numbers.

20365 **FUTURE DIRECTIONS**

20366 None.

20367 **SEE ALSO**

20368 Chapter 2 (on page 29), *ps*, *wait*, the System Interfaces volume of IEEE Std 1003.1-2001, *kill()*, the
20369 Base Definitions volume of IEEE Std 1003.1-2001, [<signal.h>](#)

20370 **CHANGE HISTORY**

20371 First released in Issue 2.

20372 **Issue 6**

20373 The obsolescent versions of the SYNOPSIS are turned into non-obsolescent features of the XSI
20374 option, corresponding to a similar change in the *trap* special built-in.

20375 NAME

20376 lex — generate programs for lexical tasks (DEVELOPMENT)

20377 SYNOPSIS

20378 CD lex [-t][-n|-v][file ...]

20379

20380 DESCRIPTION

20381 The *lex* utility shall generate C programs to be used in lexical processing of character input, and
20382 that can be used as an interface to *yacc*. The C programs shall be generated from *lex* source code
20383 and conform to the ISO C standard. Usually, the *lex* utility shall write the program it generates to
20384 the file **lex.yy.c**; the state of this file is unspecified if *lex* exits with a non-zero exit status. See the
20385 EXTENDED DESCRIPTION section for a complete description of the *lex* input language.

20386 OPTIONS

20387 The *lex* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
20388 Utility Syntax Guidelines.

20389 The following options shall be supported:

20390 **-n** Suppress the summary of statistics usually written with the **-v** option. If no table
20391 sizes are specified in the *lex* source code and the **-v** option is not specified, then **-n**
20392 is implied.

20393 **-t** Write the resulting program to standard output instead of **lex.yy.c**.

20394 **-v** Write a summary of *lex* statistics to the standard output. (See the discussion of *lex*
20395 table sizes in **Definitions in lex** (on page 534).) If the **-t** option is specified and **-n**
20396 is not specified, this report shall be written to standard error. If table sizes are
20397 specified in the *lex* source code, and if the **-n** option is not specified, the **-v** option
20398 may be enabled.

20399 OPERANDS

20400 The following operand shall be supported:

20401 *file* A pathname of an input file. If more than one such *file* is specified, all files shall be
20402 concatenated to produce a single *lex* program. If no *file* operands are specified, or if
20403 a *file* operand is '**-**', the standard input shall be used.

20404 STDIN

20405 The standard input shall be used if no *file* operands are specified, or if a *file* operand is '**-**'. See
20406 INPUT FILES.

20407 INPUT FILES

20408 The input files shall be text files containing *lex* source code, as described in the EXTENDED
20409 DESCRIPTION section.

20410 ENVIRONMENT VARIABLES

20411 The following environment variables shall affect the execution of *lex*:

20412 **LANG** Provide a default value for the internationalization variables that are unset or null.
20413 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
20414 Internationalization Variables for the precedence of internationalization variables
20415 used to determine the values of locale categories.)

20416 **LC_ALL** If set to a non-empty string value, override the values of all the other
20417 internationalization variables.

20418 **LC_COLLATE**

20419 Determine the locale for the behavior of ranges, equivalence classes, and multi-

20420	character collating elements within regular expressions. If this variable is not set to the POSIX locale, the results are unspecified.
20422	LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), and the behavior of character classes within regular expressions. If this variable is not set to the POSIX locale, the results are unspecified.
20427	LC_MESSAGES
20428	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
20429	
20430 XSI	NLSPATH Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
20431	ASYNCHRONOUS EVENTS
20432	Default.
20433	STDOUT
20434	If the -t option is specified, the text file of C source code output of <i>lex</i> shall be written to standard output.
20435	
20436	If the -t option is not specified:
20437	• Implementation-defined informational, error, and warning messages concerning the contents of <i>lex</i> source code input shall be written to either the standard output or standard error.
20438	
20439	• If the -v option is specified and the -n option is not specified, <i>lex</i> statistics shall also be written to either the standard output or standard error, in an implementation-defined format. These statistics may also be generated if table sizes are specified with a '%' operator in the <i>Definitions</i> section, as long as the -n option is not specified.
20440	
20441	
20442	
20443	STDERR
20444	If the -t option is specified, implementation-defined informational, error, and warning messages concerning the contents of <i>lex</i> source code input shall be written to the standard error.
20445	
20446	If the -t option is not specified:
20447	1. Implementation-defined informational, error, and warning messages concerning the contents of <i>lex</i> source code input shall be written to either the standard output or standard error.
20448	
20449	
20450	2. If the -v option is specified and the -n option is not specified, <i>lex</i> statistics shall also be written to either the standard output or standard error, in an implementation-defined format. These statistics may also be generated if table sizes are specified with a '%' operator in the <i>Definitions</i> section, as long as the -n option is not specified.
20451	
20452	
20453	
20454	OUTPUT FILES
20455	A text file containing C source code shall be written to lex.yy.c , or to the standard output if the -t option is present.
20456	
20457	EXTENDED DESCRIPTION
20458	Each input file shall contain <i>lex</i> source code, which is a table of regular expressions with corresponding actions in the form of C program fragments.
20459	
20460	When lex.yy.c is compiled and linked with the <i>lex</i> library (using the -l l operand with <i>c99</i>), the resulting program shall read character input from the standard input and shall partition it into strings that match the given expressions.
20461	
20462	

20463 When an expression is matched, these actions shall occur:

- 20464 • The input string that was matched shall be left in `yytext` as a null-terminated string; `yytext`
20465 shall either be an external character array or a pointer to a character string. As explained in
20466 **Definitions in lex**, the type can be explicitly selected using the `%array` or `%pointer`
20467 declarations, but the default is implementation-defined.
- 20468 • The external `int yyleng` shall be set to the length of the matching string.
- 20469 • The expression's corresponding program fragment, or action, shall be executed.

20470 During pattern matching, *lex* shall search the set of patterns for the single longest possible
20471 match. Among rules that match the same number of characters, the rule given first shall be
20472 chosen.

20473 The general format of *lex* source shall be:

```
20474 Definitions
20475 %% 
20476 Rules
20477 %%
20478 UserSubroutines
```

20479 The first "%%" is required to mark the beginning of the rules (regular expressions and actions);
20480 the second "%%" is required only if user subroutines follow.

20481 Any line in the *Definitions* section beginning with a <blank> shall be assumed to be a C program
20482 fragment and shall be copied to the external definition area of the `lex.yy.c` file. Similarly,
20483 anything in the *Definitions* section included between delimiter lines containing only "%{" and
20484 "%}" shall also be copied unchanged to the external definition area of the `lex.yy.c` file.

20485 Any such input (beginning with a <blank> or within "%{" and "%}" delimiter lines) appearing
20486 at the beginning of the *Rules* section before any rules are specified shall be written to `lex.yy.c`
20487 after the declarations of variables for the `yylex()` function and before the first line of code in
20488 `yylex()`. Thus, user variables local to `yylex()` can be declared here, as well as application code to
20489 execute upon entry to `yylex()`.

20490 The action taken by *lex* when encountering any input beginning with a <blank> or within "%{"
20491 and "%}" delimiter lines appearing in the *Rules* section but coming after one or more rules is
20492 undefined. The presence of such input may result in an erroneous definition of the `yylex()`
20493 function.

20494 **Definitions in lex**

20495 *Definitions* appear before the first "%%" delimiter. Any line in this section not contained between
20496 "%{" and "%}" lines and not beginning with a <blank> shall be assumed to define a *lex*
20497 substitution string. The format of these lines shall be:

```
20498 name substitute
```

20499 If a *name* does not meet the requirements for identifiers in the ISO C standard, the result is
20500 undefined. The string *substitute* shall replace the string {*name*} when it is used in a rule. The *name*
20501 string shall be recognized in this context only when the braces are provided and when it does
20502 not appear within a bracket expression or within double-quotes.

20503 In the *Definitions* section, any line beginning with a '%' (percent sign) character and followed by
20504 an alphanumeric word beginning with either 's' or 'S' shall define a set of start conditions.
20505 Any line beginning with a '%' followed by a word beginning with either 'x' or 'X' shall define
20506 a set of exclusive start conditions. When the generated scanner is in a %s state, patterns with no

20507 state specified shall be also active; in a %x state, such patterns shall not be active. The rest of the
 20508 line, after the first word, shall be considered to be one or more <blank>-separated names of start
 20509 conditions. Start condition names shall be constructed in the same way as definition names. Start
 20510 conditions can be used to restrict the matching of regular expressions to one or more states as
 20511 described in **Regular Expressions in lex** (on page 536).

20512 Implementations shall accept either of the following two mutually-exclusive declarations in the
 20513 *Definitions* section:

20514 **%array** Declare the type of *yytext* to be a null-terminated character array.

20515 **%pointer** Declare the type of *yytext* to be a pointer to a null-terminated character string.

20516 The default type of *yytext* is implementation-defined. If an application refers to *yytext* outside of
 20517 the scanner source file (that is, via an **extern**), the application shall include the appropriate
 20518 **%array** or **%pointer** declaration in the scanner source file.

20519 Implementations shall accept declarations in the *Definitions* section for setting certain internal
 20520 table sizes. The declarations are shown in the following table.

20521 **Table 4-9** Table Size Declarations in *lex*

Declaration	Description	Minimum Value
%p n	Number of positions	2 500
%n n	Number of states	500
%a n	Number of transitions	2 000
%e n	Number of parse tree nodes	1 000
%k n	Number of packed character classes	1 000
%o n	Size of the output array	3 000

20529 In the table, *n* represents a positive decimal integer, preceded by one or more <blank>s. The
 20530 exact meaning of these table size numbers is implementation-defined. The implementation shall
 20531 document how these numbers affect the *lex* utility and how they are related to any output that
 20532 may be generated by the implementation should limitations be encountered during the
 20533 execution of *lex*. It shall be possible to determine from this output which of the table size values
 20534 needs to be modified to permit *lex* to successfully generate tables for the input language. The
 20535 values in the column Minimum Value represent the lowest values conforming implementations
 20536 shall provide.

20537 Rules in lex

20538 The rules in *lex* source files are a table in which the left column contains regular expressions and
 20539 the right column contains actions (C program fragments) to be executed when the expressions
 20540 are recognized.

20541 *ERE action*
 20542 *ERE action*
 20543 . . .

20544 The extended regular expression (ERE) portion of a row shall be separated from *action* by one or
 20545 more <blank>s. A regular expression containing <blank>s shall be recognized under one of the
 20546 following conditions:

- 20547 • The entire expression appears within double-quotes.
- 20548 • The <blank>s appear within double-quotes or square brackets.

- 20549 • Each <blank> is preceded by a backslash character.

20550 **User Subroutines in lex**

20551 Anything in the user subroutines section shall be copied to **lex.yy.c** following **yylex()**.

20552 **Regular Expressions in lex**

20553 The *lex* utility shall support the set of extended regular expressions (see the Base Definitions
20554 volume of IEEE Std 1003.1-2001, Section 9.4, Extended Regular Expressions), with the following
20555 additions and exceptions to the syntax:

- 20556 " . . ." Any string enclosed in double-quotes shall represent the characters within the
20557 double-quotes as themselves, except that backslash escapes (which appear in the
20558 following table) shall be recognized. Any backslash-escape sequence shall be
20559 terminated by the closing quote. For example, "\01" "1" represents a single
20560 string: the octal value 1 followed by the character '1'.

20561 <state>r, <state1,state2,...>r

20562 The regular expression *r* shall be matched only when the program is in one of the
20563 start conditions indicated by *state*, *state1*, and so on; see **Actions in lex** (on page
20564 538). (As an exception to the typographical conventions of the rest of this volume
20565 of IEEE Std 1003.1-2001, in this case <state> does not represent a metavariable, but
20566 the literal angle-bracket characters surrounding a symbol.) The start condition
20567 shall be recognized as such only at the beginning of a regular expression.

- 20568 r/x The regular expression *r* shall be matched only if it is followed by an occurrence of
20569 regular expression *x* (*x* is the instance of trailing context, further defined below).
20570 The token returned in *yytext* shall only match *r*. If the trailing portion of *r* matches
20571 the beginning of *x*, the result is unspecified. The *r* expression cannot include
20572 further trailing context or the '\$' (match-end-of-line) operator; *x* cannot include
20573 the '^' (match-beginning-of-line) operator, nor trailing context, nor the '\$'
20574 operator. That is, only one occurrence of trailing context is allowed in a *lex* regular
20575 expression, and the '^' operator only can be used at the beginning of such an
20576 expression.

- 20577 {name} When *name* is one of the substitution symbols from the *Definitions* section, the
20578 string, including the enclosing braces, shall be replaced by the *substitute* value. The
20579 *substitute* value shall be treated in the extended regular expression as if it were
20580 enclosed in parentheses. No substitution shall occur if {name} occurs within a
20581 bracket expression or within double-quotes.

20582 Within an ERE, a backslash character shall be considered to begin an escape sequence as
20583 specified in the table in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File
20584 Format Notation ('\\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v'). In addition, the escape
20585 sequences in the following table shall be recognized.

20586 A literal <newline> cannot occur within an ERE; the escape sequence '\n' can be used to
20587 represent a <newline>. A <newline> shall not be matched by a period operator.

20588

Table 4-10 Escape Sequences in *lex*

20589 20590 20591 20592 20593 20594 20595 20596 20597 20598 20599 20600 20601 20602	20591 20592 20593 20594 20595 20596 20597 20598 20599 20600 20601 20602	20591 20592 20593 20594 20595 20596 20597 20598 20599 20600 20601 20602
	\digits	A backslash character followed by the longest sequence of one, two, or three octal-digit characters (01234567). If all of the digits are 0 (that is, representation of the NUL character), the behavior is undefined.
20603 20604 20605 20606 20607 20608 20609 20610	\xdigits	A backslash character followed by the longest sequence of hexadecimal-digit characters (01234567abcdefABCDEF). If all of the digits are 0 (that is, representation of the NUL character), the behavior is undefined.
20611 20612 20613 20614 20615 20616 20617 20618	\c	A backslash character followed by any character not described in this table or in the table in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation ('\\', '\\a', '\\b', '\\f', '\\n', '\\r', '\\t', '\\v').

20619 **Note:** If a '\x' sequence needs to be immediately followed by a hexadecimal digit character, a sequence such as "\x1""1" can be used, which represents a character containing the value 1, followed by the character '1'.

20622 The order of precedence given to extended regular expressions for *lex* differs from that specified in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.4, Extended Regular Expressions. The order of precedence for *lex* shall be as shown in the following table, from high to low.

20626 **Note:** The escaped characters entry is not meant to imply that these are operators, but they are included in the table to show their relationships to the true operators. The start condition, trailing context, and anchoring notations have been omitted from the table because of the placement restrictions described in this section; they can only appear at the beginning or ending of an ERE.

20631

Table 4-11 ERE Precedence in *lex*

Extended Regular Expression	Precedence
<i>collation-related bracket symbols</i>	[= =] [: :] [. .]
<i>escaped characters</i>	\<special character>
<i>bracket expression</i>	[]
<i>quoting</i>	" . . . "
<i>grouping</i>	()
<i>definition</i>	{ name }
<i>single-character RE duplication</i>	* + ?
<i>concatenation</i>	
<i>interval expression</i>	{ m , n }
<i>alternation</i>	

The ERE anchoring operators '^' and '\$' do not appear in the table. With *lex* regular expressions, these operators are restricted in their use: the '^' operator can only be used at the beginning of an entire regular expression, and the '\$' operator only at the end. The operators apply to the entire regular expression. Thus, for example, the pattern "(^abc)|(def\$)" is undefined; it can instead be written as two separate rules, one with the regular expression "^abc" and one with "def\$", which share a common action via the special '|' action (see below). If the pattern were written "^abc|def\$", it would match either "abc" or "def" on a line by itself.

Unlike the general ERE rules, embedded anchoring is not allowed by most historical *lex* implementations. An example of embedded anchoring would be for patterns such as "^(|)foo(| \$)" to match "foo" when it exists as a complete word. This functionality can be obtained using existing *lex* features:

```
20655 ^foo/[ \n]      |
20656 " foo"/[ \n]    /* Found foo as a separate word. */
```

Note also that '\$' is a form of trailing context (it is equivalent to "/\n") and as such cannot be used with regular expressions containing another instance of the operator (see the preceding discussion of trailing context).

The additional regular expressions trailing-context operator '//' can be used as an ordinary character if presented within double-quotes, "/" ; preceded by a backslash, "\/" ; or within a bracket expression, "[/]". The start-condition '<' and '>' operators shall be special only in a start condition at the beginning of a regular expression; elsewhere in the regular expression they shall be treated as ordinary characters.

Actions in lex

The action to be taken when an ERE is matched can be a C program fragment or the special actions described below; the program fragment can contain one or more C statements, and can also include special actions. The empty C statement ';' shall be a valid action; any string in the *lex.yy.c* input that matches the pattern portion of such a rule is effectively ignored or skipped. However, the absence of an action shall not be valid, and the action *lex* takes in such a condition is undefined.

The specification for an action, including C statements and special actions, can extend across several lines if enclosed in braces:

```
20674 ERE <one or more blanks> { program statement
20675                                program statement }
```

20676 The default action when a string in the input to a **lex.y.c** program is not matched by any
 20677 expression shall be to copy the string to the output. Because the default behavior of a program
 20678 generated by *lex* is to read the input and copy it to the output, a minimal *lex* source program that
 20679 has just "%%" shall generate a C program that simply copies the input to the output unchanged.

20680 Four special actions shall be available:

20681 | ECHO; REJECT; BEGIN

20682 | The action ' | ' means that the action for the next rule is the action for this rule.
 20683 Unlike the other three actions, ' | ' cannot be enclosed in braces or be semicolon-
 20684 terminated; the application shall ensure that it is specified alone, with no other
 20685 actions.

20686 **ECHO;** Write the contents of the string *yytext* on the output.

20687 **REJECT;** Usually only a single expression is matched by a given string in the input. **REJECT**
 20688 means "continue to the next expression that matches the current input", and shall
 20689 cause whatever rule was the second choice after the current rule to be executed for
 20690 the same input. Thus, multiple rules can be matched and executed for one input
 20691 string or overlapping input strings. For example, given the regular expressions
 20692 "xyz" and "xy" and the input "xyz", usually only the regular expression "xyz"
 20693 would match. The next attempted match would start after **z**. If the last action in the
 20694 "xyz" rule is **REJECT**, both this rule and the "xy" rule would be executed. The
 20695 **REJECT** action may be implemented in such a fashion that flow of control does not
 20696 continue after it, as if it were equivalent to a **goto** to another part of *yylex()*. The
 20697 use of **REJECT** may result in somewhat larger and slower scanners.

20698 **BEGIN** The action:

20699 BEGIN *newstate*;

20700 switches the state (start condition) to *newstate*. If the string *newstate* has not been
 20701 declared previously as a start condition in the *Definitions* section, the results are
 20702 unspecified. The initial state is indicated by the digit '0' or the token **INITIAL**.

20703 The functions or macros described below are accessible to user code included in the *lex* input. It
 20704 is unspecified whether they appear in the C code output of *lex*, or are accessible only through the
 20705 **-l l** operand to *c99* (the *lex* library).

20706 **int yylex(void)**

20707 Performs lexical analysis on the input; this is the primary function generated by the *lex*
 20708 utility. The function shall return zero when the end of input is reached; otherwise, it shall
 20709 return non-zero values (tokens) determined by the actions that are selected.

20710 **int yymore(void)**

20711 When called, indicates that when the next input string is recognized, it is to be appended to
 20712 the current value of *yytext* rather than replacing it; the value in *yyleng* shall be adjusted
 20713 accordingly.

20714 **int yyless(int n)**

20715 Retains *n* initial characters in *yytext*, NUL-terminated, and treats the remaining characters
 20716 as if they had not been read; the value in *yyleng* shall be adjusted accordingly.

20717 **int input(void)**

20718 Returns the next character from the input, or zero on end-of-file. It shall obtain input from
 20719 the stream pointer *yyin*, although possibly via an intermediate buffer. Thus, once scanning
 20720 has begun, the effect of altering the value of *yyin* is undefined. The character read shall be
 20721 removed from the input stream of the scanner without any processing by the scanner.

20722 **int unput(int c)**
20723 Returns the character 'c' to the input; *yytext* and *yylen* are undefined until the next
20724 expression is matched. The result of using *unput()* for more characters than have been input
20725 is unspecified.

20726 The following functions shall appear only in the *lex* library accessible through the **-l l** operand;
20727 they can therefore be redefined by a conforming application:

20728 **int yywrap(void)**
20729 Called by *yylex()* at end-of-file; the default *yywrap()* shall always return 1. If the application
20730 requires *yylex()* to continue processing with another source of input, then the application
20731 can include a function *yywrap()*, which associates another file with the external variable
20732 **FILE * yyin** and shall return a value of zero.

20733 **int main(int argc, char *argv[])**
20734 Calls *yylex()* to perform lexical analysis, then exits. The user code can contain *main()* to
20735 perform application-specific operations, calling *yylex()* as applicable.

20736 Except for *input()*, *unput()*, and *main()*, all external and static names generated by *lex* shall begin
20737 with the prefix **yy** or **YY**.

20738 EXIT STATUS

20739 The following exit values shall be returned:

20740 0 Successful completion.
20741 >0 An error occurred.

20742 CONSEQUENCES OF ERRORS

20743 Default.

20744 APPLICATION USAGE

20745 Conforming applications are warned that in the *Rules* section, an ERE without an action is not
20746 acceptable, but need not be detected as erroneous by *lex*. This may result in compilation or
20747 runtime errors.

20748 The purpose of *input()* is to take characters off the input stream and discard them as far as the
20749 lexical analysis is concerned. A common use is to discard the body of a comment once the
20750 beginning of a comment is recognized.

20751 The *lex* utility is not fully internationalized in its treatment of regular expressions in the *lex*
20752 source code or generated lexical analyzer. It would seem desirable to have the lexical analyzer
20753 interpret the regular expressions given in the *lex* source according to the environment specified
20754 when the lexical analyzer is executed, but this is not possible with the current *lex* technology.
20755 Furthermore, the very nature of the lexical analyzers produced by *lex* must be closely tied to the
20756 lexical requirements of the input language being described, which is frequently locale-specific
20757 anyway. (For example, writing an analyzer that is used for French text is not automatically
20758 useful for processing other languages.)

20759 EXAMPLES

20760 The following is an example of a *lex* program that implements a rudimentary scanner for a
20761 Pascal-like syntax:

```
20762   %{  
20763   /* Need this for the call to atof() below. */  
20764   #include <math.h>  
20765   /* Need this for printf(), fopen(), and stdin below. */  
20766   #include <stdio.h>  
20767   }%
```

```

20768     DIGIT      [0-9]
20769     ID         [a-z][a-z0-9]*
20770     %%
20771     {DIGIT}+ {
20772         printf("An integer: %s (%d)\n", yytext,
20773             atoi(yytext));
20774     }
20775     {DIGIT}+.{DIGIT}*      {
20776         printf("A float: %s (%g)\n", yytext,
20777             atof(yytext));
20778     }
20779     if|then|begin|end|procedure|function      {
20780         printf("A keyword: %s\n", yytext);
20781     }
20782     {ID}      printf("An identifier: %s\n", yytext);
20783     "+" | "-" | "*" | "/"      printf("An operator: %s\n", yytext);
20784     " {[^}\n]*}" /* Eat up one-line comments. */
20785     [ \t\n]+      /* Eat up white space. */
20786     .  printf("Unrecognized character: %s\n", yytext);
20787     %%
20788     int main(int argc, char *argv[])
20789     {
20790         ++argv, --argc; /* Skip over program name. */
20791         if (argc > 0)
20792             yyin = fopen(argv[0], "r");
20793         else
20794             yyin = stdin;
20795         yylex();
20796     }

```

20797 RATIONALE

Even though the `-c` option and references to the C language are retained in this description, *lex* may be generalized to other languages, as was done at one time for EFL, the Extended FORTRAN Language. Since the *lex* input specification is essentially language-independent, versions of this utility could be written to produce Ada, Modula-2, or Pascal code, and there are known historical implementations that do so.

The current description of *lex* bypasses the issue of dealing with internationalized EREs in the *lex* source code or generated lexical analyzer. If it follows the model used by *awk* (the source code is assumed to be presented in the POSIX locale, but input and output are in the locale specified by the environment variables), then the tables in the lexical analyzer produced by *lex* would interpret EREs specified in the *lex* source in terms of the environment variables specified when *lex* was executed. The desired effect would be to have the lexical analyzer interpret the EREs given in the *lex* source according to the environment specified when the lexical analyzer is executed, but this is not possible with the current *lex* technology.

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

20813 being represented by octal values. Hexadecimal values can represent larger bytes and multi-byte
20814 characters directly, using as many digits as required.

20815 There is no detailed output format specification. The observed behavior of *lex* under four
20816 different historical implementations was that none of these implementations consistently
20817 reported the line numbers for error and warning messages. Furthermore, there was a desire that
20818 *lex* be allowed to output additional diagnostic messages. Leaving message formats unspecified
20819 avoids these formatting questions and problems with internationalization.

20820 Although the %x specifier for *exclusive* start conditions is not historical practice, it is believed to
20821 be a minor change to historical implementations and greatly enhances the usability of *lex*
20822 programs since it permits an application to obtain the expected functionality with fewer
20823 statements.

20824 The %array and %pointer declarations were added as a compromise between historical systems.
20825 The System V-based *lex* copies the matched text to a yytext array. The *flex* program, supported in
20826 BSD and GNU systems, uses a pointer. In the latter case, significant performance improvements
20827 are available for some scanners. Most historical programs should require no change in porting
20828 from one system to another because the string being referenced is null-terminated in both cases.
20829 (The method used by *flex* in its case is to null-terminate the token in place by remembering the
20830 character that used to come right after the token and replacing it before continuing on to the next
20831 scan.) Multi-file programs with external references to yytext outside the scanner source file
20832 should continue to operate on their historical systems, but would require one of the new
20833 declarations to be considered strictly portable.

20834 The description of EREs avoids unnecessary duplication of ERE details because their meanings
20835 within a *lex* ERE are the same as that for the ERE in this volume of IEEE Std 1003.1-2001.

20836 The reason for the undefined condition associated with text beginning with a <blank> or within
20837 "%{" and "%}" delimiter lines appearing in the *Rules* section is historical practice. Both the BSD
20838 and System V *lex* copy the indented (or enclosed) input in the *Rules* section (except at the
20839 beginning) to unreachable areas of the *yylex()* function (the code is written directly after a *break*
20840 statement). In some cases, the System V *lex* generates an error message or a syntax error,
20841 depending on the form of indented input.

20842 The intention in breaking the list of functions into those that may appear in **lex.y.c** versus those
20843 that only appear in **libl.a** is that only those functions in **libl.a** can be reliably redefined by a
20844 conforming application.

20845 The descriptions of standard output and standard error are somewhat complicated because
20846 historical *lex* implementations chose to issue diagnostic messages to standard output (unless -t
20847 was given). IEEE Std 1003.1-2001 allows this behavior, but leaves an opening for the more
20848 expected behavior of using standard error for diagnostics. Also, the System V behavior of
20849 writing the statistics when any table sizes are given is allowed, while BSD-derived systems can
20850 avoid it. The programmer can always precisely obtain the desired results by using either the -t
20851 or -n options.

20852 The OPERANDS section does not mention the use of - as a synonym for standard input; not all
20853 historical implementations support such usage for any of the *file* operands.

20854 A description of the *translation table* was deleted from early proposals because of its relatively
20855 low usage in historical applications.

20856 The change to the definition of the *input()* function that allows buffering of input presents the
20857 opportunity for major performance gains in some applications.

20858 The following examples clarify the differences between *lex* regular expressions and regular
20859 expressions appearing elsewhere in this volume of IEEE Std 1003.1-2001. For regular expressions

20860 of the form "*r/x*", the string matching *r* is always returned; confusion may arise when the
20861 beginning of *x* matches the trailing portion of *r*. For example, given the regular expression
20862 "*a*b/cc*" and the input "aaabcc", yytext would contain the string "aaab" on this match. But
20863 given the regular expression "*x*/xy*" and the input "xxxy", the token **xxx**, not **xx**, is returned
20864 by some implementations because **xxx** matches "*x**".

20865 In the rule "*ab*/bc*", the "*b**" at the end of *r* extends *r*'s match into the beginning of the
20866 trailing context, so the result is unspecified. If this rule were "*ab/bc*", however, the rule
20867 matches the text "ab" when it is followed by the text "bc". In this latter case, the matching of *r*
20868 cannot extend into the beginning of *x*, so the result is specified.

20869 **FUTURE DIRECTIONS**

20870 None.

20871 **SEE ALSO**

20872 *c99, ed, yacc*

20873 **CHANGE HISTORY**

20874 First released in Issue 2.

20875 **Issue 6**

20876 This utility is marked as part of the C-Language Development Utilities option.

20877 The obsolescent **-c** option is withdrawn in this issue.

20878 The normative text is reworded to avoid use of the term "must" for application requirements.

20879 NAME

20880 link — call *link()* function

20881 SYNOPSIS

20882 XSI link *file1 file2*

20883

20884 DESCRIPTION

20885 The *link* utility shall perform the function call:

20886 link(*file1, file2*);

20887 A user may need appropriate privilege to invoke the *link* utility.

20888 OPTIONS

20889 None.

20890 OPERANDS

20891 The following operands shall be supported:

20892 *file1* The pathname of an existing file.

20893 *file2* The pathname of the new directory entry to be created.

20894 STDIN

20895 Not used.

20896 INPUT FILES

20897 Not used.

20898 ENVIRONMENT VARIABLES

20899 The following environment variables shall affect the execution of *link*:

20900 *LANG* Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
Internationalization Variables for the precedence of internationalization variables
used to determine the values of locale categories.)

20904 *LC_ALL* If set to a non-empty string value, override the values of all the other
internationalization variables.

20906 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
characters (for example, single-byte as opposed to multi-byte characters in
arguments).

20909 *LC_MESSAGES* Determine the locale that should be used to affect the format and contents of
diagnostic messages written to standard error.

20912 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

20913 ASYNCHRONOUS EVENTS

20914 Default.

20915 STDOUT

20916 None.

20917 STDERR

20918 The standard error shall be used only for diagnostic messages.

20919 OUTPUT FILES

20920 None.

20921 EXTENDED DESCRIPTION

20922 None.

20923 EXIT STATUS

20924 The following exit values shall be returned:

20925 0 Successful completion.

20926 >0 An error occurred.

20927 CONSEQUENCES OF ERRORS

20928 Default.

20929 APPLICATION USAGE

20930 None.

20931 EXAMPLES

20932 None.

20933 RATIONALE

20934 None.

20935 FUTURE DIRECTIONS

20936 None.

20937 SEE ALSO

20938 *In*, *unlink*, the System Interfaces volume of IEEE Std 1003.1-2001, *link()*

20939 CHANGE HISTORY

20940 First released in Issue 5.

20941 NAME

20942 *In* — link files

20943 SYNOPSIS

20944 *ln* [-fs] *source_file target_file*20945 *ln* [-fs] *source_file ... target_dir*

20946 DESCRIPTION

20947 In the first synopsis form, the *In* utility shall create a new directory entry (link) at the destination path specified by the *target_file* operand. If the *-s* option is specified, a symbolic link shall be created for the file specified by the *source_file* operand. This first synopsis form shall be assumed when the final operand does not name an existing directory; if more than two operands are specified and the final is not an existing directory, an error shall result.

20952 In the second synopsis form, the *In* utility shall create a new directory entry (link), or if the *-s* option is specified a symbolic link, for each file specified by a *source_file* operand, at a destination path in the existing directory named by *target_dir*.

20955 If the last operand specifies an existing file of a type not specified by the System Interfaces volume of IEEE Std 1003.1-2001, the behavior is implementation-defined.

20957 The corresponding destination path for each *source_file* shall be the concatenation of the target directory pathname, a slash character, and the last pathname component of the *source_file*. The second synopsis form shall be assumed when the final operand names an existing directory.

20960 For each *source_file*:

- 20961 1. If the destination path exists:

- 20962 a. If the *-f* option is not specified, *In* shall write a diagnostic message to standard error, do nothing more with the current *source_file*, and go on to any remaining *source_files*.

- 20964 b. Actions shall be performed equivalent to the *unlink()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001, called using *destination* as the *path* argument. If this fails for any reason, *In* shall write a diagnostic message to standard error, do nothing more with the current *source_file*, and go on to any remaining *source_files*.

- 20969 2. If the *-s* option is specified, *In* shall create a symbolic link named by the destination path and containing as its pathname *source_file*. The *In* utility shall do nothing more with *source_file* and shall go on to any remaining files.

- 20973 3. If *source_file* is a symbolic link, actions shall be performed equivalent to the *link()* function using the object that *source_file* references as the *path1* argument and the destination path as the *path2* argument. The *In* utility shall do nothing more with *source_file* and shall go on to any remaining files.

- 20977 4. Actions shall be performed equivalent to the *link()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001 using *source_file* as the *path1* argument, and the destination path as the *path2* argument.

20979 OPTIONS

20980 The *In* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

20982 The following option shall be supported:

20983 **-f** Force existing destination pathnames to be removed to allow the link.

20984 **-s** Create symbolic links instead of hard links.

20985 OPERANDS

20986 The following operands shall be supported:

20987 *source_file* A pathname of a file to be linked. If the **-s** option is specified, no restrictions on the
20988 type of file or on its existence shall be made. If the **-s** option is not specified,
20989 whether a directory can be linked is implementation-defined.

20990 *target_file* The pathname of the new directory entry to be created.

20991 *target_dir* A pathname of an existing directory in which the new directory entries are created.

20992 STDIN

20993 Not used.

20994 INPUT FILES

20995 None.

20996 ENVIRONMENT VARIABLES

20997 The following environment variables shall affect the execution of *In*:

20998 *LANG* Provide a default value for the internationalization variables that are unset or null.
20999 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
21000 Internationalization Variables for the precedence of internationalization variables
21001 used to determine the values of locale categories.)

21002 *LC_ALL* If set to a non-empty string value, override the values of all the other
21003 internationalization variables.

21004 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
21005 characters (for example, single-byte as opposed to multi-byte characters in
21006 arguments).

21007 *LC_MESSAGES*

21008 Determine the locale that should be used to affect the format and contents of
21009 diagnostic messages written to standard error.

21010 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

21011 ASYNCHRONOUS EVENTS

21012 Default.

21013 STDOUT

21014 Not used.

21015 STDERR

21016 The standard error shall be used only for diagnostic messages.

21017 OUTPUT FILES

21018 None.

21019 EXTENDED DESCRIPTION

21020 None.

21021 EXIT STATUS

21022 The following exit values shall be returned:

21023 0 All the specified files were linked successfully.

21024 >0 An error occurred.

21025 **CONSEQUENCES OF ERRORS**

21026 Default.

21027 **APPLICATION USAGE**

21028 None.

21029 **EXAMPLES**

21030 None.

21031 **RATIONALE**

21032 Some historic versions of *ln* (including the one specified by the SVID) unlink the destination file, if it exists, by default. If the mode does not permit writing, these versions prompt for confirmation before attempting the unlink. In these versions the **-f** option causes *ln* not to attempt to prompt for confirmation.

21036 This allows *ln* to succeed in creating links when the target file already exists, even if the file itself is not writable (although the directory must be). Early proposals specified this functionality.

21038 This volume of IEEE Std 1003.1-2001 does not allow the *ln* utility to unlink existing destination paths by default for the following reasons:

- 21040 • The *ln* utility has historically been used to provide locking for shell applications, a usage that is incompatible with *ln* unlinking the destination path by default. There was no corresponding technical advantage to adding this functionality.
- 21043 • This functionality gave *ln* the ability to destroy the link structure of files, which changes the historical behavior of *ln*.
- 21045 • This functionality is easily replicated with a combination of *rm* and *ln*.
- 21046 • It is not historical practice in many systems; BSD and BSD-derived systems do not support this behavior. Unfortunately, whichever behavior is selected can cause scripts written expecting the other behavior to fail.
- 21049 • It is preferable that *ln* perform in the same manner as the *link()* function, which does not permit the target to exist already.

21051 This volume of IEEE Std 1003.1-2001 retains the **-f** option to provide support for shell scripts depending on the SVID semantics. It seems likely that shell scripts would not be written to handle prompting by *ln* and would therefore have specified the **-f** option.

21054 The **-f** option is an undocumented feature of many historical versions of the *ln* utility, allowing linking to directories. These versions require modification.

21056 Early proposals of this volume of IEEE Std 1003.1-2001 also required a **-i** option, which behaved like the **-i** options in *cp* and *mv*, prompting for confirmation before unlinking existing files. This was not historical practice for the *ln* utility and has been omitted.

21059 **FUTURE DIRECTIONS**

21060 None.

21061 **SEE ALSO**21062 *chmod*, *find*, *pax*, *rm*, the System Interfaces volume of IEEE Std 1003.1-2001, *link()*, *unlink()*21063 **CHANGE HISTORY**

21064 First released in Issue 2.

21065 Issue 6

21066 The *ln* utility is updated to include symbolic link processing as defined in the IEEE P1003.2b
21067 draft standard.

21068 NAME

21069 *locale* — get locale-specific information

21070 SYNOPSIS

21071 *locale* [-a | -m]21072 *locale* [-ck] *name*...

21073 DESCRIPTION

21074 The *locale* utility shall write information about the current locale environment, or all public
21075 locales, to the standard output. For the purposes of this section, a *public locale* is one provided by
21076 the implementation that is accessible to the application.

21077 When *locale* is invoked without any arguments, it shall summarize the current locale
21078 environment for each locale category as determined by the settings of the environment variables
21079 defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 7, Locale.

21080 When invoked with operands, it shall write values that have been assigned to the keywords in
21081 the locale categories, as follows:

- 21082 • Specifying a keyword name shall select the named keyword and the category containing that
21083 keyword.
- 21084 • Specifying a category name shall select the named category and all keywords in that
21085 category.

21086 OPTIONS

21087 The *locale* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
21088 12.2, Utility Syntax Guidelines.

21089 The following options shall be supported:

21090 **-a** Write information about all available public locales. The available locales shall
21091 include **POSIX**, representing the POSIX locale. The manner in which the
21092 implementation determines what other locales are available is implementation-
21093 defined.

21094 **-c** Write the names of selected locale categories; see the STDOUT section. The **-c**
21095 option increases readability when more than one category is selected (for example,
21096 via more than one keyword name or via a category name). It is valid both with
21097 and without the **-k** option.

21098 **-k** Write the names and values of selected keywords. The implementation may omit
21099 values for some keywords; see the OPERANDS section.

21100 **-m** Write names of available charmaps; see the Base Definitions volume of
21101 IEEE Std 1003.1-2001, Section 6.1, Portable Character Set.

21102 OPERANDS

21103 The following operand shall be supported:

21104 *name* The name of a locale category as defined in the Base Definitions volume of
21105 IEEE Std 1003.1-2001, Chapter 7, Locale, the name of a keyword in a locale
21106 category, or the reserved name **charmap**. The named category or keyword shall be
21107 selected for output. If a single *name* represents both a locale category name and a
21108 keyword name in the current locale, the results are unspecified. Otherwise, both
21109 category and keyword names can be specified as *name* operands, in any sequence.
21110 It is implementation-defined whether any keyword values are written for the
21111 categories **LC_CTYPE** and **LC_COLLATE**.

21112 **STDIN**

21113 Not used.

21114 **INPUT FILES**

21115 None.

21116 **ENVIRONMENT VARIABLES**21117 The following environment variables shall affect the execution of *locale*:

21118 ***LANG*** Provide a default value for the internationalization variables that are unset or null.
 21119 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
 21120 Internationalization Variables for the precedence of internationalization variables
 21121 used to determine the values of locale categories.)

21122 ***LC_ALL*** If set to a non-empty string value, override the values of all the other
 21123 internationalization variables.

21124 ***LC_CTYPE*** Determine the locale for the interpretation of sequences of bytes of text data as
 21125 characters (for example, single-byte as opposed to multi-byte characters in
 21126 arguments and input files).

21127 ***LC_MESSAGES***

21128 Determine the locale that should be used to affect the format and contents of
 21129 diagnostic messages written to standard error.

21130 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

21131 XSI The application shall ensure that the *LANG*, *LC_**, and *NLSPATH* environment variables specify
 21132 the current locale environment to be written out; they shall be used if the **-a** option is not
 21133 specified.

21134 **ASYNCHRONOUS EVENTS**

21135 Default.

21136 **STDOUT**

21137 If *locale* is invoked without any options or operands, the names and values of the *LANG* and
 21138 *LC_** environment variables described in this volume of IEEE Std 1003.1-2001 shall be written to
 21139 the standard output, one variable per line, with *LANG* first, and each line using the following
 21140 format. Only those variables set in the environment and not overridden by *LC_ALL* shall be
 21141 written using this format:

21142 "%s=%s\n", <variable_name>, <value>

21143 The names of those *LC_** variables associated with locale categories defined in this volume of
 21144 IEEE Std 1003.1-2001 that are not set in the environment or are overridden by *LC_ALL* shall be
 21145 written in the following format:

21146 "%s=\" %s \"\n", <variable_name>, <implied value>

21147 The *<implied value>* shall be the name of the locale that has been selected for that category by the
 21148 implementation, based on the values in *LANG* and *LC_ALL*, as described in the Base Definitions
 21149 volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

21150 The *<value>* and *<implied value>* shown above shall be properly quoted for possible later reentry
 21151 to the shell. The *<value>* shall not be quoted using double-quotes (so that it can be distinguished
 21152 by the user from the *<implied value>* case, which always requires double-quotes).

21153 The *LC_ALL* variable shall be written last, using the first format shown above. If it is not set, it
 21154 shall be written as:

21155 "LC_ALL=\n"

21156 If any arguments are specified:

- 21157 1. If the **-a** option is specified, the names of all the public locales shall be written, each in the
21158 following format:

21159 "%s\n", <locale name>

- 21160 2. If the **-c** option is specified, the names of all selected categories shall be written, each in the
21161 following format:

21162 "%s\n", <category name>

21163 If keywords are also selected for writing (see following items), the category name output
21164 shall precede the keyword output for that category.

21165 If the **-c** option is not specified, the names of the categories shall not be written; only the
21166 keywords, as selected by the <name> operand, shall be written.

- 21167 3. If the **-k** option is specified, the names and values of selected keywords shall be written. If
21168 a value is non-numeric, it shall be written in the following format:

21169 "%s=%s\n", <keyword name>, <keyword value>

21170 If the keyword was **charmap**, the name of the charmap (if any) that was specified via the
21171 *localeddef -f* option when the locale was created shall be written, with the word **charmap** as
21172 <keyword name>.

21173 If a value is numeric, it shall be written in one of the following formats:

21174 "%s=%d\n", <keyword name>, <keyword value>

21175 "%s=%c%c\n", <keyword name>, <escape character>, <keyword value>

21176 "%s=%cx%x\n", <keyword name>, <escape character>, <keyword value>

21177 where the <escape character> is that identified by the **escape_char** keyword in the current
21178 locale; see the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3, Locale
21179 Definition.

21180 Compound keyword values (list entries) shall be separated in the output by semicolons.
21181 When included in keyword values, the semicolon, the double-quote, the backslash, and
21182 any control character shall be preceded (escaped) with the escape character.

- 21183 4. If the **-k** option is not specified, selected keyword values shall be written, each in the
21184 following format:

21185 "%s\n", <keyword value>

21186 If the keyword was **charmap**, the name of the charmap (if any) that was specified via the
21187 *localeddef -f* option when the locale was created shall be written.

- 21188 5. If the **-m** option is specified, then a list of all available charmaps shall be written, each in
21189 the format:

21190 "%s\n", <charmap>

21191 where <charmap> is in a format suitable for use as the option-argument to the *localeddef -f*
21192 option.

21193 STDERR

21194 The standard error shall be used only for diagnostic messages.

21195 OUTPUT FILES

21196 None.

21197 EXTENDED DESCRIPTION

21198 None.

21199 EXIT STATUS

21200 The following exit values shall be returned:

21201 0 All the requested information was found and output successfully.

21202 >0 An error occurred.

21203 CONSEQUENCES OF ERRORS

21204 Default.

21205 APPLICATION USAGE

21206 If the *LANG* environment variable is not set or set to an empty value, or one of the *LC_** environment variables is set to an unrecognized value, the actual locales assumed (if any) are implementation-defined as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

21210 Implementations are not required to write out the actual values for keywords in the categories *LC_CTYPE* and *LC_COLLATE*; however, they must write out the categories (allowing an application to determine, for example, which character classes are available).

21213 EXAMPLES

21214 In the following examples, the assumption is that locale environment variables are set as follows:

21216 *LANG=locale_x*
21217 *LC_COLLATE=locale_y*

21218 The command *locale* would result in the following output:

21219 *LANG=locale_x*
21220 *LC_CTYPE="locale_x"*
21221 *LC_COLLATE=locale_y*
21222 *LC_TIME="locale_x"*
21223 *LC_NUMERIC="locale_x"*
21224 *LC_MONETARY="locale_x"*
21225 *LC_MESSAGES="locale_x"*
21226 *LC_ALL=*

21227 The order of presentation of the categories is not specified by this volume of
21228 IEEE Std 1003.1-2001.

21229 The command:

21230 *LC_ALL=POSIX locale -ck decimal_point*

21231 would produce:

21232 *LC_NUMERIC*
21233 *decimal_point=". "*

21234 The following command shows an application of *locale* to determine whether a user-supplied
21235 response is affirmative:

```
21236     if printf "%s\n" "$response" | grep -Eq "\$(locale yesexpr)"  
21237     then  
21238         affirmative processing goes here  
21239     else  
21240         non-affirmative processing goes here  
21241     fi
```

21242 RATIONALE

21243 The output for categories *LC_CTYPE* and *LC_COLLATE* has been made implementation-defined
21244 because there is a questionable value in having a shell script receive an entire array of characters.
21245 It is also difficult to return a logical collation description, short of returning a complete *localeddef*
21246 source.

21247 The **-m** option was included to allow applications to query for the existence of charmaps. The
21248 output is a list of the charmaps (implementation-supplied and user-supplied, if any) on the
21249 system.

21250 The **-c** option was included for readability when more than one category is selected (for
21251 example, via more than one keyword name or via a category name). It is valid both with and
21252 without the **-k** option.

21253 The **charmap** keyword, which returns the name of the charmap (if any) that was used when the
21254 current locale was created, was included to allow applications needing the information to
21255 retrieve it.

21256 FUTURE DIRECTIONS

21257 None.

21258 SEE ALSO

21259 *localeddef*, the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3, Locale Definition

21260 CHANGE HISTORY

21261 First released in Issue 4.

21262 Issue 5

21263 The FUTURE DIRECTIONS section is added.

21264 Issue 6

21265 The normative text is reworded to avoid use of the term “must” for application requirements.

21266 NAME

21267 localedef — define locale environment

21268 SYNOPSIS

21269 localedef [-c][-f *charmap*][-i *sourcefile*][-u *code_set_name*] *name*

21270 DESCRIPTION

21271 The *localedef* utility shall convert source definitions for locale categories into a format usable by
21272 the functions and utilities whose operational behavior is determined by the setting of the locale
21273 environment variables defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter
21274 7, Locale. It is implementation-defined whether users have the capability to create new locales,
21275 in addition to those supplied by the implementation. If the symbolic constant
21276 **XSI** **POSIX2_LOCALEDEF** is defined, the system supports the creation of new locales. On **XSI**-
21277 conformant systems, the symbolic constant **POSIX2_LOCALEDEF** shall be defined.

21278 The utility shall read source definitions for one or more locale categories belonging to the same
21279 locale from the file named in the **-i** option (if specified) or from standard input.

21280 The *name* operand identifies the target locale. The utility shall support the creation of *public*, or
21281 generally accessible locales, as well as *private*, or restricted-access locales. Implementations may
21282 restrict the capability to create or modify public locales to users with the appropriate privileges.

21283 Each category source definition shall be identified by the corresponding environment variable
21284 name and terminated by an **END** *category-name* statement. The following categories shall be
21285 supported. In addition, the input may contain source for implementation-defined categories.

21286 **LC_CTYPE** Defines character classification and case conversion.

21287 **LC_COLLATE**

21288 Defines collation rules.

21289 **LC_MONETARY**

21290 Defines the format and symbols used in formatting monetary information.

21291 **LC_NUMERIC**

21292 Defines the decimal delimiter, grouping, and grouping symbol for non-monetary
21293 numeric editing.

21294 **LC_TIME** Defines the format and content of date and time information.

21295 **LC_MESSAGES**

21296 Defines the format and values of affirmative and negative responses.

21297 OPTIONS

21298 The *localedef* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
21299 12.2, Utility Syntax Guidelines.

21300 The following options shall be supported:

21301 **-c** Create permanent output even if warning messages have been issued.

21302 **-f** *charmap* Specify the pathname of a file containing a mapping of character symbols and
21303 collating element symbols to actual character encodings. The format of the
21304 *charmap* is described in the Base Definitions volume of IEEE Std 1003.1-2001,
21305 Section 6.4, Character Set Description File. The application shall ensure that this
21306 option is specified if symbolic names (other than collating symbols defined in a
21307 **collating-symbol** keyword) are used. If the **-f** option is not present, an
21308 implementation-defined character mapping shall be used.

- 21309 **-i *inputfile*** The pathname of a file containing the source definitions. If this option is not
 21310 present, source definitions shall be read from standard input. The format of the
 21311 *inputfile* is described in the Base Definitions volume of IEEE Std 1003.1-2001,
 21312 Section 7.3, Locale Definition.
- 21313 **-u *code_set_name***
 21314 Specify the name of a codeset used as the target mapping of character symbols and
 21315 collating element symbols whose encoding values are defined in terms of the
 21316 ISO/IEC 10646-1:2000 standard position constant values.

21317 OPERANDS

21318 The following operand shall be supported:

- 21319 ***name*** Identifies the locale; see the Base Definitions volume of IEEE Std 1003.1-2001,
 21320 Chapter 7, Locale for a description of the use of this name. If the name contains one
 21321 or more slash characters, *name* shall be interpreted as a pathname where the
 21322 created locale definitions shall be stored. If *name* does not contain any slash
 21323 characters, the interpretation of the name is implementation-defined and the locale
 21324 shall be public. This capability may be restricted to users with appropriate
 21325 privileges. (As a consequence of specifying one *name*, although several categories
 21326 can be processed in one execution, only categories belonging to the same locale can
 21327 be processed.)

21328 STDIN

21329 Unless the **-i** option is specified, the standard input shall be a text file containing one or more
 21330 locale category source definitions, as described in the Base Definitions volume of
 21331 IEEE Std 1003.1-2001, Section 7.3, Locale Definition. When lines are continued using the escape
 21332 character mechanism, there is no limit to the length of the accumulated continued line.

21333 INPUT FILES

21334 The character set mapping file specified as the **charmap** option-argument is described in the Base
 21335 Definitions volume of IEEE Std 1003.1-2001, Section 6.4, Character Set Description File. If a locale
 21336 category source definition contains a **copy** statement, as defined in the Base Definitions volume
 21337 of IEEE Std 1003.1-2001, Chapter 7, Locale, and the **copy** statement names a valid, existing locale,
 21338 then **localedef** shall behave as if the source definition had contained a valid category source
 21339 definition for the named locale.

21340 ENVIRONMENT VARIABLES

21341 The following environment variables shall affect the execution of **localedef**:

- 21342 **LANG** Provide a default value for the internationalization variables that are unset or null.
 21343 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
 21344 Internationalization Variables for the precedence of internationalization variables
 21345 used to determine the values of locale categories.)
- 21346 **LC_ALL** If set to a non-empty string value, override the values of all the other
 21347 internationalization variables.
- 21348 **LC_COLLATE**
 21349 (This variable has no affect on **localedef**; the POSIX locale is used for this category.)
- 21350 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
 21351 characters (for example, single-byte as opposed to multi-byte characters in
 21352 arguments and input files). This variable has no affect on the processing of **localedef**
 21353 input data; the POSIX locale is used for this purpose, regardless of the value of this
 21354 variable.

21355	LC_MESSAGES	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
21356		
21357		
21358 XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
21359	ASYNCHRONOUS EVENTS	
21360		Default.
21361	STDOUT	
21362		The utility shall report all categories successfully processed, in an unspecified format.
21363	STDERR	
21364		The standard error shall be used only for diagnostic messages.
21365	OUTPUT FILES	
21366		The format of the created output is unspecified. If the <i>name</i> operand does not contain a slash, the existence of an output file for the locale is unspecified.
21367		
21368	EXTENDED DESCRIPTION	
21369		When the -u option is used, the <i>code_set_name</i> option-argument shall be interpreted as an implementation-defined name of a codeset to which the ISO/IEC 10646-1:2000 standard position constant values shall be converted via an implementation-defined method. Both the ISO/IEC 10646-1:2000 standard position constant values and other formats (decimal, hexadecimal, or octal) shall be valid as encoding values within the <i>charmap</i> file. The codeset represented by the implementation-defined name can be any codeset that is supported by the implementation.
21370		
21371		
21372		
21373		
21374		
21375		
21376		When conflicts occur between the <i>charmap</i> specification of < <i>code_set_name</i> >, < <i>mb_cur_max</i> >, or < <i>mb_cur_min</i> > and the implementation-defined interpretation of these respective items for the codeset represented by the -u option-argument <i>code_set_name</i> , the result is unspecified.
21377		
21378		
21379		When conflicts occur between the <i>charmap</i> encoding values specified for symbolic names of characters of the portable character set and the implementation-defined assignment of character encoding values, the result is unspecified.
21380		
21381		
21382		If a non-printable character in the <i>charmap</i> has a width specified that is not -1 , <i>localedef</i> shall generate a warning.
21383		
21384	EXIT STATUS	
21385		The following exit values shall be returned:
21386	0	No errors occurred and the locales were successfully created.
21387	1	Warnings occurred and the locales were successfully created.
21388	2	The locale specification exceeded implementation limits or the coded character set or sets used were not supported by the implementation, and no locale was created.
21389		
21390	3	The capability to create new locales is not supported by the implementation.
21391	>3	Warnings or errors occurred and no output was created.
21392	CONSEQUENCES OF ERRORS	
21393		If an error is detected, no permanent output shall be created.
21394		If warnings occur, permanent output shall be created if the -c option was specified. The following conditions shall cause warning messages to be issued:
21395		
21396		<ul style="list-style-type: none"> • If a symbolic name not found in the <i>charmap</i> file is used for the descriptions of the <i>LC_CTYPE</i> or <i>LC_COLLATE</i> categories (for other categories, this shall be an error condition).
21397		

- 21398 • If the number of operands to the **order** keyword exceeds the {COLL_WEIGHTS_MAX} limit.
21399 • If optional keywords not supported by the implementation are present in the source.
21400 • If a non-printable character has a width specified other than -1.
21401 Other implementation-defined conditions may also cause warnings.

21402 APPLICATION USAGE

21403 The *charmap* definition is optional, and is contained outside the locale definition. This allows
21404 both completely self-defined source files, and generic sources (applicable to more than one
21405 codeset). To aid portability, all *charmap* definitions must use the same symbolic names for the
21406 portable character set. As explained in the Base Definitions volume of IEEE Std 1003.1-2001,
21407 Section 6.4, Character Set Description File, it is implementation-defined whether or not users or
21408 applications can provide additional character set description files. Therefore, the -f option might
21409 be operable only when an implementation-defined *charmap* is named.

21410 EXAMPLES

21411 None.

21412 RATIONALE

21413 The output produced by the *localedef* utility is implementation-defined. The *name* operand is
21414 used to identify the specific locale. (As a consequence, although several categories can be
21415 processed in one execution, only categories belonging to the same locale can be processed.)

21416 FUTURE DIRECTIONS

21417 None.

21418 SEE ALSO

21419 *locale*, the Base Definitions volume of IEEE Std 1003.1-2001, Section 7.3, Locale Definition

21420 CHANGE HISTORY

21421 First released in Issue 4.

21422 Issue 6

21423 The -u option is added, as specified in the IEEE P1003.2b draft standard.

21424 The normative text is reworded to avoid use of the term “must” for application requirements.

21425 NAME

21426 logger — log messages

21427 SYNOPSIS

21428 logger *string* ...

21429 DESCRIPTION

21430 The *logger* utility saves a message, in an unspecified manner and format, containing the *string* operands provided by the user. The messages are expected to be evaluated later by personnel performing system administration tasks.

21433 It is implementation-defined whether messages written in locales other than the POSIX locale are effective.

21435 OPTIONS

21436 None.

21437 OPERANDS

21438 The following operand shall be supported:

21439 *string* One of the string arguments whose contents are concatenated together, in the order specified, separated by single <space>s.

21441 STDIN

21442 Not used.

21443 INPUT FILES

21444 None.

21445 ENVIRONMENT VARIABLES

21446 The following environment variables shall affect the execution of *logger*:

21447 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

21451 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

21453 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

21456 *LC_MESSAGES*

21457 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error. (This means diagnostics from *logger* to the user or application, not diagnostic messages that the user is sending to the system administrator.)

21461 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

21462 ASYNCHRONOUS EVENTS

21463 Default.

21464 STDOUT

21465 Not used.

21466 STDERR

21467 The standard error shall be used only for diagnostic messages.

21468 OUTPUT FILES

21469 Unspecified.

21470 EXTENDED DESCRIPTION

21471 None.

21472 EXIT STATUS

21473 The following exit values shall be returned:

21474 0 Successful completion.

21475 >0 An error occurred.

21476 CONSEQUENCES OF ERRORS

21477 Default.

21478 APPLICATION USAGE

21479 This utility allows logging of information for later use by a system administrator or programmer
21480 in determining why non-interactive utilities have failed. The locations of the saved messages,
21481 their format, and retention period are all unspecified. There is no method for a conforming
21482 application to read messages, once written.

21483 EXAMPLES

21484 A batch application, running non-interactively, tries to read a configuration file and fails; it may
21485 attempt to notify the system administrator with:

21486 `logger myname: unable to read file foo. [timestamp]`

21487 RATIONALE

21488 The standard developers believed strongly that some method of alerting administrators to errors
21489 was necessary. The obvious example is a batch utility, running non-interactively, that is unable
21490 to read its configuration files or that is unable to create or write its results file. However, the
21491 standard developers did not wish to define the format or delivery mechanisms as they have
21492 historically been (and will probably continue to be) very system-specific, as well as involving
21493 functionality clearly outside the scope of this volume of IEEE Std 1003.1-2001.

21494 The text with *LC_MESSAGES* about diagnostic messages means diagnostics from *logger* to the
21495 user or application, not diagnostic messages that the user is sending to the system administrator.

21496 Multiple *string* arguments are allowed, similar to *echo*, for ease-of-use.

21497 Like the utilities *mailx* and *lp*, *logger* is admittedly difficult to test. This was not deemed sufficient
21498 justification to exclude these utilities from this volume of IEEE Std 1003.1-2001. It is also
21499 arguable that they are, in fact, testable, but that the tests themselves are not portable.

21500 FUTURE DIRECTIONS

21501 None.

21502 SEE ALSO

21503 *lp*, *mailx*, *write*

21504 CHANGE HISTORY

21505 First released in Issue 4.

21506 NAME

21507 logname — return the user's login name

21508 SYNOPSIS

21509 logname

21510 DESCRIPTION

21511 The *logname* utility shall write the user's login name to standard output. The login name shall be
21512 the string that would be returned by the *getlogin()* function defined in the System Interfaces
21513 volume of IEEE Std 1003.1-2001. Under the conditions where the *getlogin()* function would fail,
21514 the *logname* utility shall write a diagnostic message to standard error and exit with a non-zero
21515 exit status.

21516 OPTIONS

21517 None.

21518 OPERANDS

21519 None.

21520 STDIN

21521 Not used.

21522 INPUT FILES

21523 None.

21524 ENVIRONMENT VARIABLES

21525 The following environment variables shall affect the execution of *logname*:

21526 *LANG* Provide a default value for the internationalization variables that are unset or null.
21527 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
21528 Internationalization Variables for the precedence of internationalization variables
21529 used to determine the values of locale categories.)

21530 *LC_ALL* If set to a non-empty string value, override the values of all the other
21531 internationalization variables.

21532 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
21533 characters (for example, single-byte as opposed to multi-byte characters in
21534 arguments).

21535 *LC_MESSAGES*

21536 Determine the locale that should be used to affect the format and contents of
21537 diagnostic messages written to standard error.

21538 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

21539 ASYNCHRONOUS EVENTS

21540 Default.

21541 STDOUT

21542 The *logname* utility output shall be a single line consisting of the user's login name:

21543 "%s\n", <login name>

21544 STDERR

21545 The standard error shall be used only for diagnostic messages.

21546 OUTPUT FILES

21547 None.

21548 EXTENDED DESCRIPTION

21549 None.

21550 EXIT STATUS

21551 The following exit values shall be returned:

21552 0 Successful completion.

21553 >0 An error occurred.

21554 CONSEQUENCES OF ERRORS

21555 Default.

21556 APPLICATION USAGE

21557 The *logname* utility explicitly ignores the *LOGNAME* environment variable because environment changes could produce erroneous results.

21559 EXAMPLES

21560 None.

21561 RATIONALE

21562 The *passwd* file is not listed as required because the implementation may have other means of mapping login names.

21564 FUTURE DIRECTIONS

21565 None.

21566 SEE ALSO

21567 *id*, *who*, the System Interfaces volume of IEEE Std 1003.1-2001, *getlogin()*

21568 CHANGE HISTORY

21569 First released in Issue 2.

21570 NAME

21571 **lp** — send files to a printer

21572 SYNOPSIS

21573 **lp [-c][-d dest][-n copies][-msw][-o option]... [-t title][file...]**

21574 DESCRIPTION

21575 The **lp** utility shall copy the input files to an output destination in an unspecified manner. The
21576 default output destination should be to a hardcopy device, such as a printer or microfilm
21577 recorder, that produces non-volatile, human-readable documents. If such a device is not
21578 available to the application, or if the system provides no such device, the **lp** utility shall exit with
21579 a non-zero exit status.

21580 The actual writing to the output device may occur some time after the **lp** utility successfully
21581 exits. During the portion of the writing that corresponds to each input file, the implementation
21582 shall guarantee exclusive access to the device.

21583 The **lp** utility shall associate a unique *request ID* with each request.

21584 Normally, a banner page is produced to separate and identify each print job. This page may be
21585 suppressed by implementation-defined conditions, such as an operator command or one of the
21586 **-o option** values.

21587 OPTIONS

21588 The **lp** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
21589 Utility Syntax Guidelines.

21590 The following options shall be supported:

21591 **-c** Exit only after further access to any of the input files is no longer required. The
21592 application can then safely delete or modify the files without affecting the output
21593 operation. Normally, files are not copied, but are linked whenever possible. If the
21594 **-c** option is not given, then the user should be careful not to remove any of the
21595 files before the request has been printed in its entirety. It should also be noted that
21596 in the absence of the **-c** option, any changes made to the named files after the
21597 request is made but before it is printed may be reflected in the printed output. On
21598 some implementations, **-c** may be on by default.

21599 **-d dest** Specify a string that names the destination (*dest*). If *dest* is a printer, the request
21600 shall be printed only on that specific printer. If *dest* is a class of printers, the request
21601 shall be printed on the first available printer that is a member of the class. Under
21602 certain conditions (printer unavailability, file space limitation, and so on), requests
21603 for specific destinations need not be accepted. Destination names vary between
21604 systems.

21605 If **-d** is not specified, and neither the *LPDEST* nor *PRINTER* environment variable
21606 is set, an unspecified destination is used. The **-d dest** option shall take precedence
21607 over *LPDEST*, which in turn shall take precedence over *PRINTER*. Results are
21608 undefined when *dest* contains a value that is not a valid destination name.

21609 **-m** Send mail (see *mailx*) after the files have been printed. By default, no mail is sent
21610 upon normal completion of the print request.

21611 **-n copies** Write *copies* number of copies of the files, where *copies* is a positive decimal integer.
21612 The methods for producing multiple copies and for arranging the multiple copies
21613 when multiple *file* operands are used are unspecified, except that each file shall be
21614 output as an integral whole, not interleaved with portions of other files.

21615	-o <i>option</i>	Specify printer-dependent or class-dependent <i>options</i> . Several such <i>options</i> may be collected by specifying the -o option more than once.
21617	-s	Suppress messages from <i>lp</i> .
21618	-t <i>title</i>	Write <i>title</i> on the banner page of the output.
21619	-w	Write a message on the user's terminal after the files have been printed. If the user is not logged in, then mail shall be sent instead.
21620		

21621 OPERANDS

21622 The following operand shall be supported:

21623	<i>file</i>	A pathname of a file to be output. If no <i>file</i> operands are specified, or if a <i>file</i> operand is ' - ', the standard input shall be used. If a <i>file</i> operand is used, but the -c option is not specified, the process performing the writing to the output device may have user and group permissions that differ from that of the process invoking <i>lp</i> .
21624		
21625		
21626		
21627		

21628 STDIN

21629 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '**-**'.
21630 See the INPUT FILES section.

21631 INPUT FILES

21632 The input files shall be text files.

21633 ENVIRONMENT VARIABLES

21634 The following environment variables shall affect the execution of *lp*:

21635	LANG	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
21636		
21637		
21638		
21639	LC_ALL	If set to a non-empty string value, override the values of all the other internationalization variables.
21640		
21641	LC_CTYPE	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).
21642		
21643		
21644	LC_MESSAGES	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.
21645		
21646		
21647		
21648	LC_TIME	Determine the format and contents of date and time strings displayed in the <i>lp</i> banner page, if any.
21649		
21650	LPDEST	Determine the destination. If the <i>LPDEST</i> environment variable is not set, the <i>PRINTER</i> environment variable shall be used. The -d <i>dest</i> option takes precedence over <i>LPDEST</i> . Results are undefined when -d is not specified and <i>LPDEST</i> contains a value that is not a valid destination name.
21651		
21652		
21653		
21654 XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
21655		
21656		
21657		
21658	PRINTER	Determine the output device or destination. If the <i>LPDEST</i> and <i>PRINTER</i> environment variables are not set, an unspecified output device is used. The -d <i>dest</i> option and the <i>LPDEST</i> environment variable shall take precedence over <i>PRINTER</i> . Results are undefined when -d is not specified, <i>LPDEST</i> is unset, and

- 21659 *PRINTER* contains a value that is not a valid device or destination name.
- 21660 **TZ** Determine the timezone used to calculate date and time strings displayed in the *lp* banner page, if any. If *TZ* is unset or null, an unspecified default timezone shall be used.
- 21663 **ASYNCHRONOUS EVENTS**
- 21664 Default.
- 21665 **STDOUT**
- 21666 The *lp* utility shall write a *request ID* to the standard output, unless *-s* is specified. The format of the message is unspecified. The request ID can be used on systems supporting the historical *cancel* and *lpstat* utilities.
- 21669 **STDERR**
- 21670 The standard error shall be used only for diagnostic messages.
- 21671 **OUTPUT FILES**
- 21672 None.
- 21673 **EXTENDED DESCRIPTION**
- 21674 None.
- 21675 **EXIT STATUS**
- 21676 The following exit values shall be returned:
- 21677 0 All input files were processed successfully.
- 21678 >0 No output device was available, or an error occurred.
- 21679 **CONSEQUENCES OF ERRORS**
- 21680 Default.
- 21681 **APPLICATION USAGE**
- 21682 The *pr* and *fold* utilities can be used to achieve reasonable formatting for the implementation's default page size.
- 21684 A conforming application can use one of the *file* operands only with the *-c* option or if the file is publicly readable and guaranteed to be available at the time of printing. This is because IEEE Std 1003.1-2001 gives the implementation the freedom to queue up the request for printing at some later time by a different process that might not be able to access the file.
- 21688 **EXAMPLES**
- 21689 1. To print file *file*:
- 21690 *lp -c file*
- 21691 2. To print multiple files with headers:
- 21692 *pr file1 file2 | lp*
- 21693 **RATIONALE**
- 21694 The *lp* utility was designed to be a basic version of a utility that is already available in many historical implementations. The standard developers considered that it should be implementable simply as:
- 21697 *cat "\$@" > /dev/lp*
- 21698 after appropriate processing of options, if that is how the implementation chose to do it and if exclusive access could be granted (so that two users did not write to the device simultaneously).
- 21699 Although in the future the standard developers may add other options to this utility, it should

21701 always be able to execute with no options or operands and send the standard input to an
21702 unspecified output device.

21703 This volume of IEEE Std 1003.1-2001 makes no representations concerning the format of the
21704 printed output, except that it must be “human-readable” and “non-volatile”. Thus, writing by
21705 default to a disk or tape drive or a display terminal would not qualify. (Such destinations are not
21706 prohibited when **-d dest**, **LPDEST**, or **PRINTER** are used, however.)

21707 This volume of IEEE Std 1003.1-2001 is worded such that a “print job” consisting of multiple
21708 input files, possibly in multiple copies, is guaranteed to print so that any one file is not
21709 intermixed with another, but there is no statement that all the files or copies have to print out
21710 together.

21711 The **-c** option may imply a spooling operation, but this is not required. The utility can be
21712 implemented to wait until the printer is ready and then wait until it is finished. Because of that,
21713 there is no attempt to define a queuing mechanism (priorities, classes of output, and so on).

21714 On some historical systems, the request ID reported on the **STDOUT** can be used to later cancel
21715 or find the status of a request using utilities not defined in this volume of IEEE Std 1003.1-2001.

21716 Although the historical System V *lp* and BSD *lpr* utilities have provided similar functionality,
21717 they used different names for the environment variable specifying the destination printer. Since
21718 the name of the utility here is *lp*, **LPDEST** (used by the System V *lp* utility) was given precedence
21719 over **PRINTER** (used by the BSD *lpr* utility). Since environments of users frequently contain one
21720 or the other environment variable, the *lp* utility is required to recognize both. If this was not
21721 done, many applications would send output to unexpected output devices when users moved
21722 from system to system.

21723 Some have commented that *lp* has far too little functionality to make it worthwhile. Requests
21724 have proposed additional options or operands or both that added functionality. The requests
21725 included:

- 21726 • Wording *requiring* the output to be “hardcopy”
21727 • A requirement for multiple printers
21728 • Options for supporting various page-description languages

21729 Given that a compliant system is not required to even have a printer, placing further restrictions
21730 upon the behavior of the printer is not useful. Since hardcopy format is so application-
21731 dependent, it is difficult, if not impossible, to select a reasonable subset of functionality that
21732 should be required on all compliant systems.

21733 The term *unspecified* is used in this section in lieu of *implementation-defined* as most known
21734 implementations would not be able to make definitive statements in their conformance
21735 documents; the existence and usage of printers is very dependent on how the system
21736 administrator configures each individual system.

21737 Since the default destination, device type, queuing mechanisms, and acceptable forms of input
21738 are all *unspecified*, usage guidelines for what a conforming application can do are as follows:

- 21739 • Use the command in a pipeline, or with **-c**, so that there are no permission problems and the
21740 files can be safely deleted or modified.
21741 • Limit output to text files of reasonable line lengths and printable characters and include no
21742 device-specific formatting information, such as a page description language. The meaning of
21743 “reasonable” in this context can only be answered as a quality-of-implementation issue, but
21744 it should be apparent from historical usage patterns in the industry and the locale. The *pr* and
21745 *fold* utilities can be used to achieve reasonable formatting for the default page size of the

21746 implementation.

21747 Alternatively, the application can arrange its installation in such a way that it requires the
21748 system administrator or operator to provide the appropriate information on *lp* options and
21749 environment variable values.

21750 At a minimum, having this utility in this volume of IEEE Std 1003.1-2001 tells the industry that
21751 conforming applications require a means to print output and provides at least a command name
21752 and *LPDEST* routing mechanism that can be used for discussions between vendors, application
21753 writers, and users. The use of “should” in the DESCRIPTION of *lp* clearly shows the intent of
21754 the standard developers, even if they cannot mandate that all systems (such as laptops) have
21755 printers.

21756 This volume of IEEE Std 1003.1-2001 does not specify what the ownership of the process
21757 performing the writing to the output device may be. If **-c** is not used, it is unspecified whether
21758 the process performing the writing to the output device has permission to read *file* if there are
21759 any restrictions in place on who may read *file* until after it is printed. Also, if **-c** is not used, the
21760 results of deleting *file* before it is printed are unspecified.

21761 FUTURE DIRECTIONS

21762 None.

21763 SEE ALSO

21764 *mailx*

21765 CHANGE HISTORY

21766 First released in Issue 2.

21767 Issue 6

21768 The following new requirements on POSIX implementations derive from alignment with the
21769 Single UNIX Specification:

- 21770 • In the DESCRIPTION, the requirement to associate a unique request ID, and the normal
21771 generation of a banner page is added.
- 21772 • In the OPTIONS section:
 - 21773 — The **-d** *dest* description is expanded, but references to *lpstat* are removed.
 - 21774 — The **-m**, **-o**, **-s**, **-t**, and **-w** options are added.
- 21775 • In the ENVIRONMENT VARIABLES section, *LC_TIME* may now affect the execution.
- 21776 • The STDOUT section is added.

21777 The normative text is reworded to avoid use of the term “must” for application requirements.

21778 The *TZ* entry is added to the ENVIRONMENT VARIABLES section.

21779 NAME

21780 ls — list directory contents

21781 SYNOPSIS

21782 XSI ls [-CFRacdilqrtu1][-H | -L][-fgmnopsx][file...]

21783 DESCRIPTION

21784 For each operand that names a file of a type other than directory or symbolic link to a directory, *ls* shall write the name of the file as well as any requested, associated information. For each
21785 operand that names a file of type directory, *ls* shall write the names of files contained within the
21786 directory as well as any requested, associated information. If one of the **-d**, **-F**, or **-l** options are
21787 specified, and one of the **-H** or **-L** options are not specified, for each operand that names a file of
21788 type symbolic link to a directory, *ls* shall write the name of the file as well as any requested,
21789 associated information. If none of the **-d**, **-F**, or **-l** options are specified, or the **-H** or **-L** options
21790 are specified, for each operand that names a file of type symbolic link to a directory, *ls* shall write
21791 the names of files contained within the directory as well as any requested, associated
21792 information.

21793 If no operands are specified, *ls* shall write the contents of the current directory. If more than one
21794 operand is specified, *ls* shall write non-directory operands first; it shall sort directory and non-
21795 directory operands separately according to the collating sequence in the current locale.

21796 The *ls* utility shall detect infinite loops; that is, entering a previously visited directory that is an
21797 ancestor of the last file encountered. When it detects an infinite loop, *ls* shall write a diagnostic
21798 message to standard error and shall either recover its position in the hierarchy or terminate.

21800 OPTIONS

21801 The *ls* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
21802 Utility Syntax Guidelines.

21803 The following options shall be supported:

- 21804 **-C** Write multi-text-column output with entries sorted down the columns, according
21805 to the collating sequence. The number of text columns and the column separator
21806 characters are unspecified, but should be adapted to the nature of the output
21807 device.
- 21808 **-F** Do not follow symbolic links named as operands unless the **-H** or **-L** options are
21809 specified. Write a slash ('/') immediately after each pathname that is a directory,
21810 an asterisk ('*') after each that is executable, a vertical bar ('|') after each that is
21811 a FIFO, and an at sign ('@') after each that is a symbolic link. For other file types,
21812 other symbols may be written.
- 21813 **-H** If a symbolic link referencing a file of type directory is specified on the command
21814 line, *ls* shall evaluate the file information and file type to be those of the file
21815 referenced by the link, and not the link itself; however, *ls* shall write the name of
21816 the link itself and not the file referenced by the link.
- 21817 **-L** Evaluate the file information and file type for all symbolic links (whether named
21818 on the command line or encountered in a file hierarchy) to be those of the file
21819 referenced by the link, and not the link itself; however, *ls* shall write the name of
21820 the link itself and not the file referenced by the link. When **-L** is used with **-l**, write
21821 the contents of symbolic links in the long format (see the STDOUT section).
- 21822 **-R** Recursively list subdirectories encountered.
- 21823 **-a** Write out all directory entries, including those whose names begin with a period
21824 ('.'). Entries beginning with a period shall not be written out unless explicitly

21825		referenced, the -a option is supplied, or an implementation-defined condition shall cause them to be written.
21826		
21827	-c	Use time of last modification of the file status information (see <sys/stat.h> in the System Interfaces volume of IEEE Std 1003.1-2001) instead of last modification of the file itself for sorting (-t) or writing (-l).
21828		
21829		
21830	-d	Do not follow symbolic links named as operands unless the -H or -L options are specified. Do not treat directories differently than other types of files. The use of -d with -R produces unspecified results.
21831		
21832		
21833 XSI	-f	Force each argument to be interpreted as a directory and list the name found in each slot. This option shall turn off -l , -t , -s , and -r , and shall turn on -a ; the order is the order in which entries appear in the directory.
21834		
21835		
21836 XSI	-g	The same as -l , except that the owner shall not be written.
21837	-i	For each file, write the file's file serial number (see stat() in the System Interfaces volume of IEEE Std 1003.1-2001).
21838		
21839	-l	(The letter ell.) Do not follow symbolic links named as operands unless the -H or -L options are specified. Write out in long format (see the STDOUT section). When -l (ell) is specified, -1 (one) shall be assumed.
21840		
21841		
21842 XSI	-m	Stream output format; list files across the page, separated by commas.
21843 XSI	-n	The same as -l , except that the owner's UID and GID numbers shall be written, rather than the associated character strings.
21844		
21845 XSI	-o	The same as -l , except that the group shall not be written.
21846 XSI	-p	Write a slash (' / ') after each filename if that file is a directory.
21847	-q	Force each instance of non-printable filename characters and <tab> s to be written as the question-mark (' ? ') character. Implementations may provide this option by default if the output is to a terminal device.
21848		
21849		
21850	-r	Reverse the order of the sort to get reverse collating sequence or oldest first.
21851 XSI	-s	Indicate the total number of file system blocks consumed by each file displayed. The block size is implementation-defined.
21852		
21853	-t	Sort with the primary key being time modified (most recently modified first) and the secondary key being filename in the collating sequence.
21854		
21855	-u	Use time of last access (see <sys/stat.h>) instead of last modification of the file for sorting (-t) or writing (-l).
21856		
21857 XSI	-x	The same as -C , except that the multi-text-column output is produced with entries sorted across, rather than down, the columns.
21858		
21859	-1	(The numeric digit one.) Force output to be one entry per line.
21860		Specifying more than one of the options in the following mutually-exclusive pairs shall not be considered an error: -C and -l (ell), -m and -l (ell), -x and -l (ell), -C and -1 (one), -H and -L , -c and -u . The last option specified in each pair shall determine the output format.
21861 XSI		
21862		

21863 OPERANDS

21864 The following operand shall be supported:

21865 *file* A pathname of a file to be written. If the file specified is not found, a diagnostic message shall be output on standard error.

21867 **STDIN**

21868 Not used.

21869 **INPUT FILES**

21870 None.

21871 **ENVIRONMENT VARIABLES**21872 The following environment variables shall affect the execution of *ls*:

21873 **COLUMNS** Determine the user's preferred column position width for writing multiple text-column output. If this variable contains a string representing a decimal integer, the *ls* utility shall calculate how many pathname text columns to write (see **-C**) based on the width provided. If **COLUMNS** is not set or invalid, an implementation-defined number of column positions shall be assumed, based on the implementation's knowledge of the output device. The column width chosen to write the names of files in any given directory shall be constant. Filenames shall not be truncated to fit into the multiple text-column output.

21881 **LANG** Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

21885 **LC_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

21887 **LC_COLLATE** Determine the locale for character collation information in determining the pathname collation sequence.

21890 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments) and which characters are defined as printable (character class **print**).

21893 **LC_MESSAGES** Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

21896 **LC_TIME** Determine the format and contents for date and time strings written by *ls*.

21897 **XSI_NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

21898 **TZ** Determine the timezone for date and time strings written by *ls*. If **TZ** is unset or null, an unspecified default timezone shall be used.

21900 **ASYNCHRONOUS EVENTS**

21901 Default.

21902 **STDOUT**

21903 The default format shall be to list one entry per line to standard output; the exceptions are to terminals or when one of the **-C**, **-m**, or **-x** options is specified. If the output is to a terminal, the format is implementation-defined.

21906 **XSI** When **-m** is specified, the format used shall be:

21907 "%s, %s, ... \n", <filename1>, <filename2>

21908 where the largest number of filenames shall be written without exceeding the length of the line.

21909 If the **-i** option is specified, the file's file serial number (see **<sys/stat.h>**) shall be written in the following format before any other output for the corresponding entry:

21911 %u ", <file serial number>

21912 If the **-l** option is specified without **-L**, the following information shall be written:

21913 "%s %u %s %s %u %s %s\n", <file mode>, <number of links>,
 21914 <owner name>, <group name>, <number of bytes in the file>,
 21915 <date and time>, <pathname>

21916 If the file is a symbolic link, this information shall be about the link itself and the <pathname>
 21917 field shall be of the form:

21918 "%s -> %s", <pathname of link>, <contents of link>

21919 If both **-l** and **-L** are specified, the following information shall be written:

21920 "%s %u %s %s %u %s %s\n", <file mode>, <number of links>,
 21921 <owner name>, <group name>, <number of bytes in the file>,
 21922 <date and time>, <pathname of link>

21923 where all fields except <pathname of link> shall be for the file resolved from the symbolic link.

21924 XSI The **-g**, **-n**, and **-o** options use the same format as **-l**, but with omitted items and their
 21925 associated <blank>s. See the OPTIONS section.

21926 XSI In both the preceding **-l** forms, if <owner name> or <group name> cannot be determined, or if **-n**
 21927 is given, they shall be replaced with their associated numeric values using the format %u.

21928 The <date and time> field shall contain the appropriate date and timestamp of when the file was
 21929 last modified. In the POSIX locale, the field shall be the equivalent of the output of the following
 21930 *date* command:

21931 date "+%b %e %H:%M"

21932 if the file has been modified in the last six months, or:

21933 date "+%b %e %Y"

21934 (where two <space>s are used between %e and %Y) if the file has not been modified in the last six
 21935 months or if the modification date is in the future, except that, in both cases, the final <newline>
 21936 produced by *date* shall not be included and the output shall be as if the *date* command were
 21937 executed at the time of the last modification date of the file rather than the current time. When
 21938 the *LC_TIME* locale category is not set to the POSIX locale, a different format and order of
 21939 presentation of this field may be used.

21940 If the file is a character special or block special file, the size of the file may be replaced with
 21941 implementation-defined information associated with the device in question.

21942 If the pathname was specified as a *file* operand, it shall be written as specified.

21943 XSI The file mode written under the **-l**, **-g**, **-n**, and **-o** options shall consist of the following format:

21944 "%c%s%s%s%c", <entry type>, <owner permissions>,
 21945 <group permissions>, <other permissions>,
 21946 <optional alternate access method flag>

21947 The <optional alternate access method flag> shall be a single <space> if there is no alternate or
 21948 additional access control method associated with the file; otherwise, a printable character shall
 21949 be used.

21950 The <entry type> character shall describe the type of file, as follows:

21951 d Directory.

21952 b Block special file.
21953 c Character special file.
21954 l (ell) Symbolic link.
21955 p FIFO.
21956 - Regular file.

21957 Implementations may add other characters to this list to represent other implementation-defined
21958 file types.

21959 The next three fields shall be three characters each:

21960 <*owner permissions*>
21961 Permissions for the file owner class (see the Base Definitions volume of
21962 IEEE Std 1003.1-2001, Section 4.4, File Access Permissions).

21963 <*group permissions*>
21964 Permissions for the file group class.

21965 <*other permissions*>
21966 Permissions for the file other class.

21967 Each field shall have three character positions:

- 21968 1. If 'r', the file is readable; if '−', the file is not readable.
- 21969 2. If 'w', the file is writable; if '−', the file is not writable.
- 21970 3. The first of the following that applies:

21971 S If in <*owner permissions*>, the file is not executable and set-user-ID mode is set. If in
21972 <*group permissions*>, the file is not executable and set-group-ID mode is set.
21973 s If in <*owner permissions*>, the file is executable and set-user-ID mode is set. If in
21974 <*group permissions*>, the file is executable and set-group-ID mode is set.

21975 XSI T If in <*other permissions*> and the file is a directory, search permission is not granted to
21976 others, and the restricted deletion flag is set.

21977 XSI t If in <*other permissions*> and the file is a directory, search permission is granted to
21978 others, and the restricted deletion flag is set.

21979 x The file is executable or the directory is searchable.

21980 − None of the attributes of 'S', 's', 'T', 't', or 'x' applies.

21981 Implementations may add other characters to this list for the third character position. Such
21982 additions shall, however, be written in lowercase if the file is executable or searchable, and
21983 in uppercase if it is not.

21984 XSI If any of the **-l**, **-g**, **-n**, **-o**, or **-s** options is specified, each list of files within the directory shall be
21985 preceded by a status line indicating the number of file system blocks occupied by files in the
21986 directory in 512-byte units, rounded up to the next integral number of units, if necessary. In the
21987 POSIX locale, the format shall be:

21988 "total %u\n", <number of units in the directory>

21989 If more than one directory, or a combination of non-directory files and directories are written,
21990 either as a result of specifying multiple operands, or the **-R** option, each list of files within a
21991 directory shall be preceded by:

21992 "\\n%*s*:\\n", <directory name>
21993 If this string is the first thing to be written, the first <newline> shall not be written. This output
21994 shall precede the number of units in the directory.
21995 XSI If the **-s** option is given, each file shall be written with the number of blocks used by the file.
21996 Along with **-C**, **-1**, **-m**, or **-x**, the number and a <space> shall precede the filename; with **-g**, **-l**,
21997 **-n**, or **-o**, they shall precede each line describing a file.

21998 **STDERR**
21999 The standard error shall be used only for diagnostic messages.

22000 **OUTPUT FILES**
22001 None.

22002 **EXTENDED DESCRIPTION**
22003 None.

22004 **EXIT STATUS**
22005 The following exit values shall be returned:
22006 0 Successful completion.
22007 >0 An error occurred.

22008 **CONSEQUENCES OF ERRORS**
22009 Default.

22010 **APPLICATION USAGE**
22011 Many implementations use the equal sign ('=') to denote sockets bound to the file system for
22012 the **-F** option. Similarly, many historical implementations use the 's' character to denote
22013 sockets as the entry type characters for the **-I** option.
22014 It is difficult for an application to use every part of the file modes field of *ls -l* in a portable
22015 manner. Certain file types and executable bits are not guaranteed to be exactly as shown, as
22016 implementations may have extensions. Applications can use this field to pass directly to a user
22017 printout or prompt, but actions based on its contents should generally be deferred, instead, to
22018 the *test* utility.
22019 The output of *ls* (with the **-l** and related options) contains information that logically could be
22020 used by utilities such as *chmod* and *touch* to restore files to a known state. However, this
22021 information is presented in a format that cannot be used directly by those utilities or be easily
22022 translated into a format that can be used. A character has been added to the end of the
22023 permissions string so that applications at least have an indication that they may be working in
22024 an area they do not understand instead of assuming that they can translate the permissions
22025 string into something that can be used. Future issues or related documents may define one or
22026 more specific characters to be used based on different standard additional or alternative access
22027 control mechanisms.
22028 As with many of the utilities that deal with filenames, the output of *ls* for multiple files or in one
22029 of the long listing formats must be used carefully on systems where filenames can contain
22030 embedded white space. Systems and system administrators should institute policies and user
22031 training to limit the use of such filenames.
22032 The number of disk blocks occupied by the file that it reports varies depending on underlying
22033 file system type, block size units reported, and the method of calculating the number of blocks.
22034 On some file system types, the number is the actual number of blocks occupied by the file
22035 (counting indirect blocks and ignoring holes in the file); on others it is calculated based on the
22036 file size (usually making an allowance for indirect blocks, but ignoring holes).

22037 EXAMPLES

22038 An example of a small directory tree being fully listed with *ls -laRF a* in the POSIX locale:

```
22039      total 11
22040      drwxr-xr-x  3 hlj      prog        64 Jul  4 12:07 .
22041      drwxrwxrwx  4 hlj      prog        3264 Jul  4 12:09 ..
22042      drwxr-xr-x  2 hlj      prog        48 Jul  4 12:07 b/
22043      -rwxr--r--  1 hlj      prog        572 Jul  4 12:07 foo*
22044      a/b:
22045      total 4
22046      drwxr-xr-x  2 hlj      prog        48 Jul  4 12:07 .
22047      drwxr-xr-x  3 hlj      prog        64 Jul  4 12:07 ..
22048      -rw-r--r--  1 hlj      prog        700 Jul  4 12:07 bar
```

22049 RATIONALE

22050 Some historical implementations of the *ls* utility show all entries in a directory except dot and
 22051 dot-dot when a superuser invokes *ls* without specifying the **-a** option. When “normal” users
 22052 invoke *ls* without specifying **-a**, they should not see information about any files with names
 22053 beginning with a period unless they were named as *file* operands.

22054 Implementations are expected to traverse arbitrary depths when processing the **-R** option. The
 22055 only limitation on depth should be based on running out of physical storage for keeping track of
 22056 untraversed directories.

22057 The **-1** (one) option was historically found in BSD and BSD-derived implementations only. It is
 22058 required in this volume of IEEE Std 1003.1-2001 so that conforming applications might ensure
 22059 that output is one entry per line, even if the output is to a terminal.

22060 Generally, this volume of IEEE Std 1003.1-2001 is silent about what happens when options are
 22061 given multiple times. In the cases of **-C**, **-l**, and **-1**, however, it does specify the results of these
 22062 overlapping options. Since *ls* is one of the most aliased commands, it is important that the
 22063 implementation perform intuitively. For example, if the alias were:

```
22064 alias ls="ls -C"
```

22065 and the user typed *ls -1*, single-text-column output should result, not an error.

22066 The BSD *ls* provides a **-A** option (like **-a**, but dot and dot-dot are not written out). The small
 22067 difference from **-a** did not seem important enough to require both.

22068 Implementations may make **-q** the default for terminals to prevent trojan horse attacks on
 22069 terminals with special escape sequences. This is not required because:

- 22070 • Some control characters may be useful on some terminals; for example, a system might write
 22071 them as "\001" or "^A".
- 22072 • Special behavior for terminals is not relevant to applications portability.

22073 An early proposal specified that the optional alternate access method flag had to be '+' if there
 22074 was an alternate access method used on the file or <space> if there was not. This was changed to
 22075 be <space> if there is not and a single printable character if there is. This was done for three
 22076 reasons:

- 22077 1. There are historical implementations using characters other than '+'.
- 22078 2. There are implementations that vary this character used in that position to distinguish
 22079 between various alternate access methods in use.

22080 3. The standard developers did not want to preclude future specifications that might need a
22081 way to specify more than one alternate access method.

22082 Nonetheless, implementations providing a single alternate access method are encouraged to use
22083 ' +' .

22084 In an early proposal, the units used to specify the number of blocks occupied by files in a
22085 directory in an *ls -l* listing were implementation-defined. This was because BSD systems have
22086 historically used 1 024-byte units and System V systems have historically used 512-byte units. It
22087 was pointed out by BSD developers that their system has used 512-byte units in some places and
22088 1 024-byte units in other places. (System V has consistently used 512.) Therefore, this volume of
22089 IEEE Std 1003.1-2001 usually specifies 512. Future releases of BSD are expected to consistently
22090 provide 512 bytes as a default with a way of specifying 1 024-byte units where appropriate.

22091 The <date and time> field in the *-l* format is specified only for the POSIX locale. As noted, the
22092 format can be different in other locales. No mechanism for defining this is present in this volume
22093 of IEEE Std 1003.1-2001, as the appropriate vehicle is a messaging system; that is, the format
22094 should be specified as a "message".

22095 FUTURE DIRECTIONS

22096 The *-s* uses implementation-defined units and cannot be used portably; it may be withdrawn in
22097 a future version.

22098 SEE ALSO

22099 *chmod*, *find*, the System Interfaces volume of IEEE Std 1003.1-2001, *stat()*, the Base Definitions
22100 volume of IEEE Std 1003.1-2001, <**sys/stat.h**>

22101 CHANGE HISTORY

22102 First released in Issue 2.

22103 Issue 5

22104 A second FUTURE DIRECTION is added.

22105 Issue 6

22106 The following new requirements on POSIX implementations derive from alignment with the
22107 Single UNIX Specification:

- 22108 • In the *-F* option, other symbols are allowed for other file types.

22109 Treatment of symbolic links is added, as defined in the IEEE P1003.2b draft standard.

22110 The Open Group Base Resolution bwg2001-010 is applied, adding the *T* and *t* fields as an XSI
22111 extension.

22112 NAME

22113 m4 — macro processor (**DEVELOPMENT**)

22114 SYNOPSIS

22115 XSI m4 [-s][-D name[=val]]...[-U name]... file...

22116

22117 DESCRIPTION

22118 The *m4* utility is a macro processor that shall read one or more text files, process them according
22119 to their included macro statements, and write the results to standard output.

22120 OPTIONS

22121 The *m4* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
22122 Utility Syntax Guidelines, except that the order of the **-D** and **-U** options shall be significant.

22123 The following options shall be supported:

22124 **-s** Enable line synchronization output for the *c99* preprocessor phase (that is, #line
22125 directives).

22126 **-D name[=val]** Define *name* to *val* or to null if =*val* is omitted.

22128 **-U name** Undefine *name*.

22129 OPERANDS

22130 The following operand shall be supported:

22131 *file* A pathname of a text file to be processed. If no *file* is given, or if it is ‘-’, the
22132 standard input shall be read.

22133 STDIN

22134 The standard input shall be a text file that is used if no *file* operand is given, or if it is ‘-’.

22135 INPUT FILES

22136 The input file named by the *file* operand shall be a text file.

22137 ENVIRONMENT VARIABLES

22138 The following environment variables shall affect the execution of *m4*:

22139 **LANG** Provide a default value for the internationalization variables that are unset or null.
22140 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
22141 Internationalization Variables for the precedence of internationalization variables
22142 used to determine the values of locale categories.)

22143 **LC_ALL** If set to a non-empty string value, override the values of all the other
22144 internationalization variables.

22145 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
22146 characters (for example, single-byte as opposed to multi-byte characters in
22147 arguments and input files).

22148 **LC_MESSAGES**

22149 Determine the locale that should be used to affect the format and contents of
22150 diagnostic messages written to standard error.

22151 **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

22152 ASYNCHRONOUS EVENTS

22153 Default.

22154 STDOUT

22155 The standard output shall be the same as the input files, after being processed for macro
22156 expansion.

22157 STDERR

22158 The standard error shall be used to display strings with the **errprint** macro, macro tracing
22159 enabled by the **traceon** macro, the defined text for macros written by the **dumpdef** macro, or for
22160 diagnostic messages.

22161 OUTPUT FILES

22162 None.

22163 EXTENDED DESCRIPTION

22164 The *m4* utility shall compare each token from the input against the set of built-in and user-
22165 defined macros. If the token matches the name of a macro, then the token shall be replaced by
22166 the macro's defining text, if any, and rescanned for matching macro names. Once no portion of
22167 the token matches the name of a macro, it shall be written to standard output. Macros may have
22168 arguments, in which case the arguments shall be substituted into the defining text before it is
22169 rescanned.

22170 Macro calls have the form:

22171 *name*(*arg1*, *arg2*, ..., *argn*)22172 Macro names shall consist of letters, digits, and underscores, where the first character is not a
22173 digit. Tokens not of this form shall not be treated as macros.22174 The application shall ensure that the left parenthesis immediately follows the name of the
22175 macro. If a token matching the name of a macro is not followed by a left parenthesis, it is
22176 handled as a use of that macro without arguments.22177 If a macro name is followed by a left parenthesis, its arguments are the comma-separated tokens
22178 between the left parenthesis and the matching right parenthesis. Unquoted <blank>s and
22179 <newline>s preceding each argument shall be ignored. All other characters, including trailing
22180 <blank>s and <newline>s, are retained. Commas enclosed between left and right parenthesis
22181 characters do not delimit arguments.22182 Arguments are positionally defined and referenced. The string "\$1" in the defining text shall be
22183 replaced by the first argument. Systems shall support at least nine arguments; only the first nine
22184 can be referenced, using the strings "\$1" to "\$9", inclusive. The string "\$0" is replaced with
22185 the name of the macro. The string "\$#" is replaced by the number of arguments as a string. The
22186 string "\$*" is replaced by a list of all of the arguments, separated by commas. The string "\$@"
22187 is replaced by a list of all of the arguments separated by commas, and each argument is quoted
22188 using the current left and right quoting strings.22189 If fewer arguments are supplied than are in the macro definition, the omitted arguments are
22190 taken to be null. It is not an error if more arguments are supplied than are in the macro
22191 definition.22192 No special meaning is given to any characters enclosed between matching left and right quoting
22193 strings, but the quoting strings are themselves discarded. By default, the left quoting string
22194 consists of a grave accent ('') and the right quoting string consists of an acute accent ('');
22195 see also the **changequote** macro.22196 Comments are written but not scanned for matching macro names; by default, the begin-
22197 comment string consists of the number sign character and the end-comment string consists of a

22198	<newline>. See also the changeom and dnl macros.
22199	The <i>m4</i> utility shall make available the following built-in macros. They can be redefined, but once this is done the original meaning is lost. Their values shall be null unless otherwise stated.
22200	In the descriptions below, the term <i>defining text</i> refers to the value of the macro: the second argument to the define macro, among other things. Except for the first argument to the eval macro, all numeric arguments to built-in macros shall be interpreted as decimal values. The string values produced as the defining text of the decr , divnum , incr , index , len , and sysval built-in macros shall be in the form of a decimal-constant as defined in the C language.
22201	
22202	
22203	
22204	
22205	
22206	changeom The changeom macro shall set the begin-comment and end-comment strings.
22207	With no arguments, the comment mechanism shall be disabled. With a single argument, that argument shall become the begin-comment string and the <newline> shall become the end-comment string. With two arguments, the first argument shall become the begin-comment string and the second argument shall become the end-comment string. Systems shall support comment strings of at least five characters.
22208	
22209	
22210	
22211	
22212	
22213	changequote The changequote macro shall set the begin-quote and end-quote strings. With no arguments, the quote strings shall be set to the default values (that is, ``'). With a single argument, that argument shall become the begin-quote string and the <newline> shall become the end-quote string. With two arguments, the first argument shall become the begin-quote string and the second argument shall become the end-quote string. Systems shall support quote strings of at least five characters.
22214	
22215	
22216	
22217	
22218	
22219	
22220	decr The defining text of the decr macro shall be its first argument decremented by 1. It shall be an error to specify an argument containing any non-numeric characters.
22221	
22222	define The second argument shall become the defining text of the macro whose name is the first argument.
22223	
22224	defn The defining text of the defn macro shall be the quoted definition (using the current quoting strings) of its arguments.
22225	
22226	divert The <i>m4</i> utility maintains nine temporary buffers, numbered 1 to 9, inclusive. When the last of the input has been processed, any output that has been placed in these buffers shall be written to standard output in buffer-numerical order. The divert macro shall divert future output to the buffer specified by its argument. Specifying no argument or an argument of 0 shall resume the normal output process. Output diverted to a stream other than 0 to 9 shall be discarded. It shall be an error to specify an argument containing any non-numeric characters.
22227	
22228	
22229	
22230	
22231	
22232	
22233	divnum The defining text of the divnum macro shall be the number of the current output stream as a string.
22234	
22235	dnl The dnl macro shall cause <i>m4</i> to discard all input characters up to and including the next <newline>.
22236	
22237	dumpdef The dumpdef macro shall write the defined text to standard error for each of the macros specified as arguments, or, if no arguments are specified, for all macros.
22238	
22239	errprint The errprint macro shall write its arguments to standard error.
22240	eval The eval macro shall evaluate its first argument as an arithmetic expression, using 32-bit signed integer arithmetic. All of the C-language operators shall be supported, except for:
22241	
22242	

```

22243      [ ]
22244      ->
22245      ++
22246      --
22247      ( type )
22248      unary *
22249      sizeof
22250      ,
22251      .
22252      ?: 
22253      unary &

```

and all assignment operators. It shall be an error to specify any of these operators. Precedence and associativity shall be as in the ISO C standard. Systems shall support octal and hexadecimal numbers as in the ISO C standard. The second argument, if specified, shall set the radix for the result; the default is 10. The third argument, if specified, sets the minimum number of digits in the result. It shall be an error to specify the second or third argument containing any non-numeric characters.

ifdef If the first argument to the **ifdef** macro is defined, the defining text shall be the second argument. Otherwise, the defining text shall be the third argument, if specified, or the null string, if not.

ifelse The **ifelse** macro takes three or more arguments. If the first two arguments compare as equal strings (after macro expansion of both arguments), the defining text shall be the third argument. If the first two arguments do not compare as equal strings and there are three arguments, the defining text shall be null. If the first two arguments do not compare as equal strings and there are four or five arguments, the defining text shall be the fourth argument. If the first two arguments do not compare as equal strings and there are six or more arguments, the first three arguments shall be discarded and processing shall restart with the remaining arguments.

include The defining text for the **include** macro shall be the contents of the file named by the first argument. It shall be an error if the file cannot be read.

incr The defining text of the **incr** macro shall be its first argument incremented by 1. It shall be an error to specify an argument containing any non-numeric characters.

index The defining text of the **index** macro shall be the first character position (as a string) in the first argument where a string matching the second argument begins (zero origin), or -1 if the second argument does not occur.

len The defining text of the **len** macro shall be the length (as a string) of the first argument.

m4exit Exit from the *m4* utility. If the first argument is specified, it is the exit code. The default is zero. It shall be an error to specify an argument containing any non-numeric characters.

m4wrap The first argument shall be processed when EOF is reached. If the **m4wrap** macro is used multiple times, the arguments specified shall be processed in the order in which the **m4wrap** macros were processed.

maketemp The defining text shall be the first argument, with any trailing 'x' characters replaced with the current process ID as a string.

22290	popdef	The popdef macro shall delete the current definition of its arguments, replacing that definition with the previous one. If there is no previous definition, the macro is undefined.
22291		
22292		
22293	pushdef	The pushdef macro shall be equivalent to the define macro with the exception that it shall preserve any current definition for future retrieval using the popdef macro.
22294		
22295	shift	The defining text for the shift macro shall be all of its arguments except for the first one.
22296		
22297	sinclude	The sinclude macro shall be equivalent to the include macro, except that it shall not be an error if the file is inaccessible.
22298		
22299	substr	The defining text for the substr macro shall be the substring of the first argument beginning at the zero-offset character position specified by the second argument. The third argument, if specified, shall be the number of characters to select; if not specified, the characters from the starting point to the end of the first argument shall become the defining text. It shall not be an error to specify a starting point beyond the end of the first argument and the defining text shall be null. It shall be an error to specify an argument containing any non-numeric characters.
22300		
22301		
22302		
22303		
22304		
22305		
22306	syscmd	The syscmd macro shall interpret its first argument as a shell command line. The defining text shall be the string result of that command. No output redirection shall be performed by the <i>m4</i> utility. The exit status value from the command can be retrieved using the sysval macro.
22307		
22308		
22309		
22310	sysval	The defining text of the sysval macro shall be the exit value of the utility last invoked by the syscmd macro (as a string).
22311		
22312	traceon	The traceon macro shall enable tracing for the macros specified as arguments, or, if no arguments are specified, for all macros. The trace output shall be written to standard error in an unspecified format.
22313		
22314		
22315	traceoff	The traceoff macro shall disable tracing for the macros specified as arguments, or, if no arguments are specified, for all macros.
22316		
22317	translit	The defining text of the translit macro shall be the first argument with every character that occurs in the second argument replaced with the corresponding character from the third argument.
22318		
22319		
22320	undefine	The undefine macro shall delete all definitions (including those preserved using the pushdef macro) of the macros named by its arguments.
22321		
22322	undivert	The undivert macro shall cause immediate output of any text in temporary buffers named as arguments, or all temporary buffers if no arguments are specified. Buffers can be undiverted into other temporary buffers. Undiverting shall discard the contents of the temporary buffer. It shall be an error to specify an argument containing any non-numeric characters.
22323		
22324		
22325		
22326		
22327	EXIT STATUS	
22328		The following exit values shall be returned:
22329	0	Successful completion.
22330	>0	An error occurred
22331		If the m4exit macro is used, the exit value can be specified by the input file.

22332 CONSEQUENCES OF ERRORS

22333 Default.

22334 APPLICATION USAGE

22335 The **defn** macro is useful for renaming macros, especially built-ins.

22336 EXAMPLES

22337 An example of a single *m4* input file capable of generating two output files follows. The file **file1.m4** could contain lines such as:

```
22339 if(VER, 1, do_something)
22340 if(VER, 2, do_something)
```

22341 The makefile for the program might include:

```
22342 file1.1.c : file1.m4
22343         m4 -D VER=1 file1.m4 > file1.1.c
22344         ...
22345 file1.2.c : file1.m4
22346         m4 -D VER=2 file1.m4 > file1.2.c
22347         ...
```

22348 The **-U** option can be used to undefine **VER**. If **file1.m4** contains:

```
22349 if(VER, 1, do_something)
22350 if(VER, 2, do_something)
22351 ifndef(VER, do_something)
```

22352 then the makefile would contain:

```
22353 file1.0.c : file1.m4
22354         m4 -U VER file1.m4 > file1.0.c
22355         ...
22356 file1.1.c : file1.m4
22357         m4 -D VER=1 file1.m4 > file1.1.c
22358         ...
22359 file1.2.c : file1.m4
22360         m4 -D VER=2 file1.m4 > file1.2.c
22361         ...
```

22362 RATIONALE

22363 None.

22364 FUTURE DIRECTIONS

22365 None.

22366 SEE ALSO

22367 *c99*

22368 CHANGE HISTORY

22369 First released in Issue 2.

22370 Issue 5

22371 The phrase “the defined text for macros written by the **dumpdef** macro” is added to the
22372 description of **STDERR**, and the description of **dumpdef** is updated to indicate that output is
22373 written to standard error. The description of **eval** is updated to indicate that the list of excluded
22374 C operators excludes unary ‘&’ and ‘.’. In the description of **ifdef**, the phrase “and it is not
22375 defined to be zero” is deleted.

22376 **Issue 6**

- 22377 In the EXTENDED DESCRIPTION, the **eval** text is updated to include a '&' character in the excepted list.
- 22378
- 22379 The EXTENDED DESCRIPTION of **divert** is updated to clarify that there are only nine diversion buffers.
- 22380
- 22381 The normative text is reworded to avoid use of the term "must" for application requirements.
- 22382 The Open Group Base Resolution bwg2000-006 is applied.

22383 **NAME**

22384 mailx — process messages

22385 **SYNOPSIS**22386 **Send Mode**22387 mailx [-s *subject*] *address...*22388 **Receive Mode**

22389 mailx -e

22390 mailx [-HiNn][-F][-u *user*]22391 mailx -f[-HiNn][-F][*file*]22392 **DESCRIPTION**

22393 The *mailx* utility provides a message sending and receiving facility. It has two major modes,
22394 selected by the options used: Send Mode and Receive Mode.

22395 On systems that do not support the User Portability Utilities option, an application using *mailx*
22396 shall have the ability to send messages in an unspecified manner (Send Mode). Unless the first
22397 character of one or more lines is tilde ('~'), all characters in the input message shall appear in
22398 the delivered message, but additional characters may be inserted in the message before it is
22399 retrieved.

22400 On systems supporting the User Portability Utilities option, mail-receiving capabilities and other
22401 interactive features, Receive Mode, described below, also shall be enabled.

22402 **Send Mode**

22403 Send Mode can be used by applications or users to send messages from the text in standard
22404 input.

22405 **Receive Mode**

22406 Receive Mode is more oriented towards interactive users. Mail can be read and sent in this
22407 interactive mode.

22408 When reading mail, *mailx* provides commands to facilitate saving, deleting, and responding to
22409 messages. When sending mail, *mailx* allows editing, reviewing, and other modification of the
22410 message as it is entered.

22411 Incoming mail shall be stored in one or more unspecified locations for each user, collectively
22412 called the system *mailbox* for that user. When *mailx* is invoked in Receive Mode, the system
22413 *mailbox* shall be the default place to find new mail. As messages are read, they shall be marked
22414 to be moved to a secondary file for storage, unless specific action is taken. This secondary file is
22415 called the **mbox** and is normally located in the directory referred to by the *HOME* environment
22416 variable (see *MBOX* in the ENVIRONMENT VARIABLES section for a description of this file).
22417 Messages shall remain in this file until explicitly removed. When the -f option is used to read
22418 mail messages from secondary files, messages shall be retained in those files unless specifically
22419 removed. All three of these locations—system mailbox, **mbox**, and secondary file—are referred
22420 to in this section as simply “mailboxes”, unless more specific identification is required.

22421 **OPTIONS**

- 22422 The *mailx* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
22423 12.2, Utility Syntax Guidelines.
- 22424 The following options shall be supported. (Only the **-s** *subject* option shall be required on all
22425 systems. The other options are required only on systems supporting the User Portability Utilities
22426 option.)
- 22427 **-e** Test for the presence of mail in the system mailbox. The *mailx* utility shall write
22428 nothing and exit with a successful return code if there is mail to read.
- 22429 **-f** Read messages from the file named by the *file* operand instead of the system
22430 mailbox. (See also **folder**.) If no *file* operand is specified, read messages from **mbox**
22431 instead of the system mailbox.
- 22432 **-F** Record the message in a file named after the first recipient. The name is the login-
22433 name portion of the address found first on the **To:** line in the mail header.
22434 Overrides the **record** variable, if set (see **Internal Variables in mailx** (on page
22435 590).)
- 22436 **-H** Write a header summary only.
- 22437 **-i** Ignore interrupts. (See also **ignore**.)
- 22438 **-n** Do not initialize from the system default start-up file. See the EXTENDED
22439 DESCRIPTION section.
- 22440 **-N** Do not write an initial header summary.
- 22441 **-s** *subject* Set the **Subject** header field to *subject*. All characters in the *subject* string shall
22442 appear in the delivered message. The results are unspecified if *subject* is longer
22443 than {LINE_MAX} – 10 bytes or contains a <newline>.
- 22444 **-u** *user* Read the system mailbox of the login name *user*. This shall only be successful if
22445 the invoking user has the appropriate privileges to read the system mailbox of that
22446 user.

22447 **OPERANDS**

- 22448 The following operands shall be supported:

- 22449 **address** Addressee of message. When **-n** is specified and no user start-up files are accessed
22450 (see the EXTENDED DESCRIPTION section), the user or application shall ensure
22451 this is an address to pass to the mail delivery system. Any system or user start-up
22452 files may enable aliases (see **alias** under **Commands in mailx** (on page 593)) that
22453 may modify the form of **address** before it is passed to the mail delivery system.
- 22454 **file** A pathname of a file to be read instead of the system mailbox when **-f** is specified.
22455 The meaning of the *file* option-argument shall be affected by the contents of the
22456 **folder** internal variable; see **Internal Variables in mailx** (on page 590).

22457 **STDIN**

- 22458 When *mailx* is invoked in Send Mode (the first synopsis line), standard input shall be the
22459 message to be delivered to the specified addresses. When in Receive Mode, user commands shall
22460 be accepted from *stdin*. If the User Portability Utilities option is not supported, standard input
22461 lines beginning with a tilde ('~') character produce unspecified results.

- 22462 If the User Portability Utilities option is supported, then in both Send and Receive Modes,
22463 standard input lines beginning with the escape character (usually tilde ('~')) shall affect
22464 processing as described in **Command Escapes in mailx** (on page 601).

22465 INPUT FILES

22466 When *mailx* is used as described by this volume of IEEE Std 1003.1-2001, the *file* option-
22467 argument (see the **-f** option) and the **mbox** shall be text files containing mail messages,
22468 formatted as described in the OUTPUT FILES section. The nature of the system mailbox is
22469 unspecified; it need not be a file.

22470 ENVIRONMENT VARIABLES

22471 The following environment variables shall affect the execution of *mailx*:

22472 DEAD	Determine the pathname of the file in which to save partial messages in case of 22473 interrupts or delivery errors. The default shall be dead.letter in the directory 22474 named by the HOME variable. The behavior of <i>mailx</i> in saving partial messages is 22475 unspecified if the User Portability Utilities option is not supported and DEAD is 22476 not defined with the value /dev/null .
22477 EDITOR	Determine the name of a utility to invoke when the edit (see Commands in mailx 22478 (on page 593)) or ~e (see Command Escapes in mailx (on page 601)) command is 22479 XSI used. The default editor is unspecified. On XSI-conformant systems it is ed . The 22480 effects of this variable are unspecified if the User Portability Utilities option is not 22481 supported.
22482 HOME	Determine the pathname of the user's home directory.
22483 LANG	Provide a default value for the internationalization variables that are unset or null. 22484 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, 22485 Internationalization Variables for the precedence of internationalization variables 22486 used to determine the values of locale categories.)
22487 LC_ALL	If set to a non-empty string value, override the values of all the other 22488 internationalization variables.
22489 LC_CTYPE	Determine the locale for the interpretation of sequences of bytes of text data as 22490 characters (for example, single-byte as opposed to multi-byte characters in 22491 arguments and input files) and the handling of case-insensitive address and 22492 header-field comparisons.
22493 LC_TIME	Determine the format and contents of the date and time strings written by <i>mailx</i> .
22494 LC_MESSAGES	Determine the locale that should be used to affect the format and contents of 22495 diagnostic messages written to standard error and informative messages written to 22496 standard output.
22498 LISTER	Determine a string representing the command for writing the contents of the 22499 folder directory to standard output when the folders command is given (see 22500 folders in Commands in mailx (on page 593)). Any string acceptable as a 22501 command_string operand to the sh -c command shall be valid. If this variable is null 22502 or not set, the output command shall be ls . The effects of this variable are 22503 unspecified if the User Portability Utilities option is not supported.
22504 MAILRC	Determine the pathname of the start-up file. The default shall be .mailrc in the 22505 directory referred to by the HOME environment variable. The behavior of <i>mailx</i> is 22506 unspecified if the User Portability Utilities option is not supported and MAILRC is 22507 not defined with the value /dev/null .
22508 MBOX	Determine a pathname of the file to save messages from the system mailbox that 22509 have been read. The exit command shall override this function, as shall saving the 22510 message explicitly in another file. The default shall be mbox in the directory

22511	named by the <i>HOME</i> variable. The effects of this variable are unspecified if the User Portability Utilities option is not supported.
22512	
22513 XSI	NLSPATH Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
22514	
22515	PAGER Determine a string representing an output filtering or pagination command for writing the output to the terminal. Any string acceptable as a <i>command_string</i> operand to the <i>sh -c</i> command shall be valid. When standard output is a terminal device, the message output shall be piped through the command if the <i>mailx</i> internal variable <i>crt</i> is set to a value less than the number of lines in the message; see Internal Variables in mailx (on page 590). If the <i>PAGER</i> variable is null or not set, the paginator shall be either <i>more</i> or another paginator utility documented in the system documentation. The effects of this variable are unspecified if the User Portability Utilities option is not supported.
22516	
22517	
22518	
22519	
22520	
22521	
22522	
22523	SHELL Determine the name of a preferred command interpreter. The default shall be <i>sh</i> . The effects of this variable are unspecified if the User Portability Utilities option is not supported.
22524	
22525	
22526	TERM If the internal variable screen is not specified, determine the name of the terminal type to indicate in an unspecified manner the number of lines in a screenful of headers. If <i>TERM</i> is not set or is set to null, an unspecified default terminal type shall be used and the value of a screenful is unspecified. The effects of this variable are unspecified if the User Portability Utilities option is not supported.
22527	
22528	
22529	
22530	
22531	TZ This variable may determine the timezone used to calculate date and time strings written by <i>mailx</i> . If <i>TZ</i> is unset or null, an unspecified default timezone shall be used.
22532	
22533	
22534	VISUAL Determine a pathname of a utility to invoke when the visual command (see Commands in mailx (on page 593)) or ~v command-escape (see Command Escapes in mailx (on page 601)) is used. If this variable is null or not set, the full-screen editor shall be <i>vi</i> . The effects of this variable are unspecified if the User Portability Utilities option is not supported.
22535	
22536	
22537	
22538	

22539 ASYNCHRONOUS EVENTS

22540 When *mailx* is in Send Mode and standard input is not a terminal, it shall take the standard
 22541 action for all signals.

22542 In Receive Mode, or in Send Mode when standard input is a terminal, if a SIGINT signal is
 22543 received:

- 22544 1. If in command mode, the current command, if there is one, shall be aborted, and a
 22545 command-mode prompt shall be written.
- 22546 2. If in input mode:
 - 22547 a. If **ignore** is set, *mailx* shall write "@\n", discard the current input line, and continue
 22548 processing, bypassing the message-abort mechanism described in item 2b.
 - 22549 b. If the interrupt was received while sending mail, either when in Receive Mode or in
 22550 Send Mode, a message shall be written, and another subsequent interrupt, with no
 22551 other intervening characters typed, shall be required to abort the mail message. If in
 22552 Receive Mode and another interrupt is received, a command-mode prompt shall be
 22553 written. If in Send Mode and another interrupt is received, *mailx* shall terminate with
 22554 a non-zero status.

22555 In both cases listed in item b, if the message is not empty:

- 22556 i. If **save** is enabled and the file named by *DEAD* can be created, the message
22557 shall be written to the file named by *DEAD*. If the file exists, the message shall
22558 be written to replace the contents of the file.
22559 ii. If **save** is not enabled, or the file named by *DEAD* cannot be created, the
2260 message shall not be saved.

22561 The *mailx* utility shall take the standard action for all other signals.

22562 STDOUT

22563 In command and input modes, all output, including prompts and messages, shall be written to
22564 standard output.

22565 STDERR

22566 The standard error shall be used only for diagnostic messages.

22567 OUTPUT FILES

22568 Various *mailx* commands and command escapes can create or add to files, including the **mbox**,
22569 the dead-letter file, and secondary mailboxes. When *mailx* is used as described in this volume of
22570 IEEE Std 1003.1-2001, these files shall be text files, formatted as follows:

```
22571     line beginning with From<space>
22572     [one or more header-lines; see Commands in mailx (on page 593)]
22573     empty line
22574     [zero or more body lines
22575     empty line]
22576     [line beginning with From<space>...]
```

22577 where each message begins with the **From <space>** line shown, preceded by the beginning of
22578 the file or an empty line. (The **From <space>** line is considered to be part of the message header,
22579 but not one of the header-lines referred to in **Commands in mailx** (on page 593); thus, it shall not
22580 be affected by the **discard**, **ignore**, or **retain** commands.) The formats of the remainder of the
22581 **From <space>** line and any additional header lines are unspecified, except that none shall be
22582 empty. The format of a message body line is also unspecified, except that no line following an
22583 empty line shall start with **From <space>**; *mailx* shall modify any such user-entered message
22584 body lines (following an empty line and beginning with **From <space>**) by adding one or more
22585 characters to precede the '**F**'; it may add these characters to **From <space>** lines that are not
22586 preceded by an empty line.

22587 When a message from the system mailbox or entered by the user is not a text file, it is
22588 implementation-defined how such a message is stored in files written by *mailx*.

22589 EXTENDED DESCRIPTION

22590 The entire EXTENDED DESCRIPTION section shall apply only to implementations supporting
22591 the User Portability Utilities option.

22592 The *mailx* utility cannot guarantee support for all character encodings in all circumstances. For
22593 example, inter-system mail may be restricted to 7-bit data by the underlying network, 8-bit data
22594 need not be portable to non-internationalized systems, and so on. Under these circumstances, it
22595 is recommended that only characters defined in the ISO/IEC 646: 1991 standard International
22596 Reference Version (equivalent to ASCII) 7-bit range of characters be used.

22597 When *mailx* is invoked using one of the Receive Mode synopsis forms, it shall write a page of
22598 header-summary lines (if **-N** was not specified and there are messages, see below), followed by
22599 a prompt indicating that *mailx* can accept regular commands (see **Commands in mailx** (on page
22600 593)); this is termed *command mode*. The page of header-summary lines shall contain the first
22601 new message if there are new messages, or the first unread message if there are unread
22602 messages, or the first message. When *mailx* is invoked using the Send Mode synopsis and

standard input is a terminal, if no subject is specified on the command line and the **asksub** variable is set, a prompt for the subject shall be written. At this point, *mailx* shall be in input mode. This input mode shall also be entered when using one of the Receive Mode synopsis forms and a reply or new message is composed using the **reply**, **Reply**, **followup**, **Followup**, or **mail** commands and standard input is a terminal. When the message is typed and the end of the message is encountered, the message shall be passed to the mail delivery software. Commands can be entered by beginning a line with the escape character (by default, tilde ('~')) followed by a single command letter and optional arguments. See **Commands in mailx** (on page 593) for a summary of these commands. It is unspecified what effect these commands will have if standard input is not a terminal when a message is entered using either the Send Mode synopsis, or the Read Mode commands **reply**, **Reply**, **followup**, **Followup**, or **mail**.

Note: For notational convenience, this section uses the default escape character, tilde, in all references and examples.

At any time, the behavior of *mailx* shall be governed by a set of environmental and internal variables. These are flags and valued parameters that can be set and cleared via the *mailx* **set** and **unset** commands.

Regular commands are of the form:

[*command*] [*msglist*] [*argument* ...]

If no *command* is specified in command mode, **next** shall be assumed. In input mode, commands shall be recognized by the escape character, and lines not treated as commands shall be taken as input for the message.

In command mode, each message shall be assigned a sequential number, starting with 1.

All messages have a state that shall affect how they are displayed in the header summary and how they are retained or deleted upon termination of *mailx*. There is at any time the notion of a *current* message, which shall be marked by a '>' at the beginning of a line in the header summary. When *mailx* is invoked using one of the Receive Mode synopsis forms, the current message shall be the first new message, if there is a new message, or the first unread message if there is an unread message, or the first message if there are any messages, or unspecified if there are no messages in the mailbox. Each command that takes an optional list of messages (*msglist*) or an optional single message (*message*) on which to operate shall leave the current message set to the highest-numbered message of the messages specified, unless the command deletes messages, in which case the current message shall be set to the first undeleted message (that is, a message not in the deleted state) after the highest-numbered message deleted by the command, if one exists, or the first undeleted message before the highest-numbered message deleted by the command, if one exists, or to an unspecified value if there are no remaining undeleted messages. All messages shall be in one of the following states:

- | | |
|--|--|
| 22639
22640
22641 | <p>new The message is present in the system mailbox and has not been viewed by the user or moved to any other state. Messages in state <i>new</i> when <i>mailx</i> quits shall be retained in the system mailbox.</p> |
| 22642
22643
22644 | <p>unread The message has been present in the system mailbox for more than one invocation of <i>mailx</i> and has not been viewed by the user or moved to any other state. Messages in state <i>unread</i> when <i>mailx</i> quits shall be retained in the system mailbox.</p> |
| 22645
22646
22647
22648
22649
22650 | <p>read The message has been processed by one of the following commands: ~f, ~m, ~F, ~M, copy, mbox, next, pipe, print, Print, top, type, Type, undelete. The delete, dp, and dt commands may also cause the next message to be marked as <i>read</i>, depending on the value of the autoprint variable. Messages that are in the system mailbox and in state <i>read</i> when <i>mailx</i> quits shall be saved in the mbox, unless the internal variable hold was set. Messages that are in the mbox or in a secondary mailbox and in state</p> |

22651		<i>read</i> when <i>mailx</i> quits shall be retained in their current location.
22652	<i>deleted</i>	The message has been processed by one of the following commands: delete , dp , dt . Messages in state <i>deleted</i> when <i>mailx</i> quits shall be deleted. Deleted messages shall be ignored until <i>mailx</i> quits or changes mailboxes or they are specified to the undelete command; for example, the message specification <i>/string</i> shall only search the subject lines of messages that have not yet been deleted, unless the command operating on the list of messages is undelete . No deleted message or deleted message header shall be displayed by any <i>mailx</i> command other than undelete .
22660	<i>preserved</i>	The message has been processed by a preserve command. When <i>mailx</i> quits, the message shall be retained in its current location.
22662	<i>saved</i>	The message has been processed by one of the following commands: save or write . If the current mailbox is the system mailbox, and the internal variable keepsave is set, messages in the state <i>saved</i> shall be saved to the file designated by the MBOX variable (see the ENVIRONMENT VARIABLES section). If the current mailbox is the system mailbox, messages in the state <i>saved</i> shall be deleted from the current mailbox, when the quit or file command is used to exit the current mailbox.
22669		The header-summary line for each message shall indicate the state of the message.
22670		Many commands take an optional list of messages (<i>msglist</i>) on which to operate, which defaults to the current message. A <i>msglist</i> is a list of message specifications separated by <blank>s, which can include:
22673	n	Message number <i>n</i> .
22674	+	The next undeleted message, or the next deleted message for the undelete command.
22675	-	The next previous undeleted message, or the next previous deleted message for the undelete command.
22677	.	The current message.
22678	^	The first undeleted message, or the first deleted message for the undelete command.
22679	\$	The last message.
22680	*	All messages.
22681	<i>n-m</i>	An inclusive range of message numbers.
22682	<i>address</i>	All messages from <i>address</i> ; any address as shown in a header summary shall be matchable in this form.
22684	<i>/string</i>	All messages with <i>string</i> in the subject line (case ignored).
22685	:c	All messages of type <i>c</i> , where <i>c</i> shall be one of:
22686	d	Deleted messages.
22687	n	New messages.
22688	o	Old messages (any not in state <i>read</i> or <i>new</i>).
22689	r	Read messages.
22690	u	Unread messages.

22691 Other commands take an optional message (*message*) on which to operate, which defaults to the
22692 current message. All of the forms allowed for *msglist* are also allowed for *message*, but if more
22693 than one message is specified, only the first shall be operated on.

22694 Other arguments are usually arbitrary strings whose usage depends on the command involved.

22695 Start-Up in mailx

22696 At start-up time, *mailx* shall take the following steps in sequence:

- 22697 1. Establish all variables at their stated default values.
- 22698 2. Process command line options, overriding corresponding default values.
- 22699 3. Import any of the *DEAD*, *EDITOR*, *MBOX*, *LISTER*, *PAGER*, *SHELL*, or *VISUAL* variables
22700 that are present in the environment, overriding the corresponding default values.
- 22701 4. Read *mailx* commands from an unspecified system start-up file, unless the **-n** option is
22702 given, to initialize any internal *mailx* variables and aliases.
- 22703 5. Process the start-up file of *mailx* commands named in the user *MAILRC* variable.

22704 Most regular *mailx* commands are valid inside start-up files, the most common use being to set
22705 up initial display options and alias lists. The following commands shall be invalid in the start-up
22706 file: **!**, **edit**, **hold**, **mail**, **preserve**, **reply**, **Reply**, **shell**, **visual**, **Copy**, **followup**, and **Followup**.
22707 Any errors in the start-up file shall either cause *mailx* to terminate with a diagnostic message and
22708 a non-zero status or to continue after writing a diagnostic message, ignoring the remainder of
22709 the lines in the start-up file.

22710 A blank line in a start-up file shall be ignored.

22711 Internal Variables in mailx

22712 The following variables are internal *mailx* variables. Each internal variable can be set via the
22713 *mailx set* command at any time. The **unset** and **set no name** commands can be used to erase
22714 variables.

22715 In the following list, variables shown as:

22716 *variable*

22717 represent Boolean values. Variables shown as:

22718 *variable=value*

22719 shall be assigned string or numeric values. For string values, the rules in **Commands in mailx**
22720 (on page 593) concerning filenames and quoting shall also apply.

22721 The defaults specified here may be changed by the implementation-defined system start-up file
22722 unless the user specifies the **-n** option.

22723 **allnet** All network names whose login name components match shall be treated as
22724 identical. This shall cause the *msglist* message specifications to behave similarly.
22725 The default shall be **noallnet**. See also the **alternates** command and the **metoo**
22726 variable.

22727 **append** Append messages to the end of the **mbox** file upon termination instead of placing
22728 them at the beginning. The default shall be **noappend**. This variable shall not
22729 affect the **save** command when saving to **mbox**.

22730 **ask**, **asksub** Prompt for a subject line on outgoing mail if one is not specified on the command
22731 line with the **-s** option. The **ask** and **asksub** forms are synonyms; the system shall

22732		refer to asksub and noasksub in its messages, but shall accept ask and noask as user input to mean asksub and noasksub . It shall not be possible to set both ask and noasksub , or noask and asksub . The default shall be asksub , but no prompting shall be done if standard input is not a terminal.
22736	askbcc	Prompt for the blind copy list. The default shall be noaskbcc .
22737	askcc	Prompt for the copy list. The default shall be noaskcc .
22738	autoprint	Enable automatic writing of messages after delete and undelete commands. The default shall be noautoprint .
22740	bang	Enable the special-case treatment of exclamation marks ('!') in escape command lines; see the escape command and Command Escapes in mailx (on page 601). The default shall be nobang , disabling the expansion of '!' in the <i>command</i> argument to the ~! command and the ~<!command escape.
22744	cmd=command	Set the default command to be invoked by the pipe command. The default shall be nocmd .
22747	crt=number	Pipe messages having more than <i>number</i> lines through the command specified by the value of the <i>PAGER</i> variable. The default shall be nocrt . If it is set to null, the value used is implementation-defined.
22750 XSI	debug	Enable verbose diagnostics for debugging. Messages are not delivered. The default shall be nodebug .
22752	dot	When dot is set, a period on a line by itself during message input from a terminal shall also signify end-of-file (in addition to normal end-of-file). The default shall be nodot . If ignoreeof is set (see below), a setting of nodot shall be ignored and the period is the only method to terminate input mode.
22756	escape=c	Set the command escape character to be the character 'c'. By default, the command escape character shall be tilde. If escape is unset, tilde shall be used; if it is set to null, command escaping shall be disabled.
22759	flipr	Reverse the meanings of the R and r commands. The default shall be noflipr .
22760	folder=directory	The default directory for saving mail files. User-specified filenames beginning with a plus sign ('+') shall be expanded by preceding the filename with this directory name to obtain the real pathname. If <i>directory</i> does not start with a slash ('/'), the contents of <i>HOME</i> shall be prefixed to it. The default shall be nofolder . If folder is unset or set to null, user-specified filenames beginning with '+' shall refer to files in the current directory that begin with the literal '+' character. See also outfolder below. The folder value need not affect the processing of the files named in <i>MBOX</i> and <i>DEAD</i> .
22769	header	Enable writing of the header summary when entering <i>mailx</i> in Receive Mode. The default shall be header .
22771	hold	Preserve all messages that are read in the system mailbox instead of putting them in the mbox save file. The default shall be nohold .
22773	ignore	Ignore interrupts while entering messages. The default shall be noignore .
22774	ignoreeof	Ignore normal end-of-file during message input. Input can be terminated only by entering a period ('.') on a line by itself or by the ~. command escape. The default shall be noignoreeof . See also dot above.

22777	indentprefix=string	A string that shall be added as a prefix to each line that is inserted into the message by the ~m command escape. This variable shall default to one <tab>.
22780	keep	When a system mailbox, secondary mailbox, or mbox is empty, truncate it to zero length instead of removing it. The default shall be nokeep .
22782	keepsave	Keep the messages that have been saved from the system mailbox into other files in the file designated by the variable MBOX , instead of deleting them. The default shall be nokeepsave .
22785	metoo	Suppress the deletion of the login name of the user from the recipient list when replying to a message or sending to a group. The default shall be nometoo .
22787 XSI	onehop	When responding to a message that was originally sent to several recipients, the other recipient addresses are normally forced to be relative to the originating author's machine for the response. This flag disables alteration of the recipients' addresses, improving efficiency in a network where all machines can send directly to all other machines (that is, one hop away). The default shall be noonehop .
22792	outfolder	Cause the files used to record outgoing messages to be located in the directory specified by the folder variable unless the pathname is absolute. The default shall be nooutfolder . See the record variable.
22795	page	Insert a <form-feed> after each message sent through the pipe created by the pipe command. The default shall be nopage .
22797	prompt=string	Set the command-mode prompt to <i>string</i> . If <i>string</i> is null or if noprompt is set, no prompting shall occur. The default shall be to prompt with the string "? ".
22800	quiet	Refrain from writing the opening message and version when entering mailx . The default shall be noquiet .
22802	record=file	Record all outgoing mail in the file with the pathname <i>file</i> . The default shall be norecord . See also outfolder above.
22804	save	Enable saving of messages in the dead-letter file on interrupt or delivery error. See the variable DEAD for the location of the dead-letter file. The default shall be save .
22806	screen=number	Set the number of lines in a screenful of headers for the headers and z commands. If screen is not specified, a value based on the terminal type identified by the TERM environment variable, the window size, the baud rate, or some combination of these shall be used.
22811	sendwait	Wait for the background mailer to finish before returning. The default shall be nosendwait .
22813	showto	When the sender of the message was the user who is invoking mailx , write the information from the To: line instead of the From: line in the header summary. The default shall be noshowto .
22816	sign=string	Set the variable inserted into the text of a message when the ~a command escape is given. The default shall be nosign . The character sequences ' \t ' and ' \n ' shall be recognized in the variable as <tab>s and <newline>s, respectively. (See also ~i in Command Escapes in mailx (on page 601).)
22820	Sign=string	Set the variable inserted into the text of a message when the ~A command escape is given. The default shall be noSign . The character sequences ' \t ' and ' \n ' shall

22822 be recognized in the variable as <tab>s and <newline>s, respectively.

22823 **toplines=number**
22824 Set the number of lines of the message to write with the **top** command. The default
22825 shall be 5.

22826 **Commands in mailx**

22827 The following *mailx* commands shall be provided. In the following list, header refers to lines
22828 from the message header, as shown in the OUTPUT FILES section. Header-line refers to lines
22829 within the header that begin with one or more non-white-space characters, immediately
22830 followed by a colon and white space and continuing until the next line beginning with a non-
22831 white-space character or an empty line. Header-field refers to the portion of a header line prior
22832 to the first colon in that line.

22833 For each of the commands listed below, the command can be entered as the abbreviation (those
22834 characters in the Synopsis command word preceding the '['), the full command (all characters
22835 shown for the command word, omitting the '[' and ']'), or any truncation of the full
22836 command down to the abbreviation. For example, the **exit** command (shown as **ex[it]** in the
22837 Synopsis) can be entered as **ex**, **exi**, or **exit**.

22838 The arguments to commands can be quoted, using the following methods:

- 22839 • An argument can be enclosed between paired double-quotes (" ") or single-quotes (' '); any
22840 white space, shell word expansion, or backslash characters within the quotes shall be treated
22841 literally as part of the argument. A double-quote shall be treated literally within single-
22842 quotes and *vice versa*. These special properties of the quote marks shall occur only when they
22843 are paired at the beginning and end of the argument.
- 22844 • A backslash outside of the enclosing quotes shall be discarded and the following character
22845 treated literally as part of the argument.
- 22846 • An unquoted backslash at the end of a command line shall be discarded and the next line
22847 shall continue the command.

22848 Filenames, where expected, shall be subjected to the process of shell word expansions (see
22849 Section 2.6 (on page 36)); if more than a single pathname results and the command is expecting
22850 one file, the effects are unspecified. If the filename begins with an unquoted plus sign, it shall not
22851 be expanded, but treated as the named file (less the leading plus) in the **folder** directory. (See the
22852 **folder** variable.)

22853 **Declare Aliases**

22854 *Synopsis:* a[lias] [alias [address...]]
22855 g[roup] [alias [address...]]

22856 Add the given addresses to the alias specified by *alias*. The names shall be substituted when
22857 *alias* is used as a recipient address specified by the user in an outgoing message (that is, other
22858 recipients addressed indirectly through the **reply** command shall not be substituted in this
22859 manner). Mail address alias substitution shall apply only when the alias string is used as a full
22860 address; for example, when **hlj** is an alias, **hlj@posix.com** does not trigger the alias substitution. If
22861 no arguments are given, write a listing of the current aliases to standard output. If only an *alias*
22862 argument is given, write a listing of the specified alias to standard output. These listings need
22863 not reflect the same order of addresses that were entered.

22864 **Declare Alternatives**

22865 *Synopsis:* alt[ernates] name...

22866 (See also the **metoo** command.) Declare a list of alternative names for the user's login. When
22867 responding to a message, these names shall be removed from the list of recipients for the
22868 response. The comparison of names shall be in a case-insensitive manner. With no arguments,
22869 **alternates** shall write the current list of alternative names.

22870 **Change Current Directory**

22871 *Synopsis:* cd [directory]
22872 ch[dir] [directory]

22873 Change directory. If *directory* is not specified, the contents of *HOME* shall be used.

22874 **Copy Messages**

22875 *Synopsis:* c[opy] [file]
22876 c[opy] [msglist] file
22877 C[opy] [msglist]

22878 Copy messages to the file named by the pathname *file* without marking the messages as saved.
22879 Otherwise, it shall be equivalent to the **save** command.

22880 In the capitalized form, save the specified messages in a file whose name is derived from the
22881 author of the message to be saved, without marking the messages as saved. Otherwise, it shall
22882 be equivalent to the **Save** command.

22883 **Delete Messages**

22884 *Synopsis:* d[elete] [msglist]

22885 Mark messages for deletion from the mailbox. The deletions shall not occur until *mailx* quits (see
22886 the **quit** command) or changes mailboxes (see the **folder** command). If **autoprint** is set and there
22887 are messages remaining after the **delete** command, the current message shall be written as
22888 described for the **print** command (see the **print** command); otherwise, the *mailx* prompt shall be
22889 written.

22890 **Discard Header Fields**

22891 *Synopsis:* di[scard] [header-field...]
22892 ig[nore] [header-field...]

22893 Suppress the specified header fields when writing messages. Specified *header-fields* shall be
22894 added to the list of suppressed header fields. Examples of header fields to ignore are **status** and
22895 **cc**. The fields shall be included when the message is saved. The **Print** and **Type** commands shall
22896 override this command. The comparison of header fields shall be in a case-insensitive manner. If
22897 no arguments are specified, write a list of the currently suppressed header fields to standard
22898 output; the listing need not reflect the same order of header fields that were entered.

22899 If both **retain** and **discard** commands are given, **discard** commands shall be ignored.

22900 **Delete Messages and Display**

22901 *Synopsis:* dp [msglist]
22902 dt [msglist]

22903 Delete the specified messages as described for the **delete** command, except that the **autoprint**
22904 variable shall have no effect, and the current message shall be written only if it was set to a
22905 message after the last message deleted by the command. Otherwise, an informational message
22906 to the effect that there are no further messages in the mailbox shall be written, followed by the
22907 **mailx** prompt.

22908 **Echo a String**

22909 *Synopsis:* ec[ho] string ...
22910 Echo the given strings, equivalent to the shell **echo** utility.

22911 **Edit Messages**

22912 *Synopsis:* e[dit] [msglist]

22913 Edit the given messages. The messages shall be placed in a temporary file and the utility named
22914 by the **EDITOR** variable is invoked to edit each file in sequence. The default **EDITOR** is
22915 unspecified.

22916 The **edit** command does not modify the contents of those messages in the mailbox.

22917 **Exit**

22918 *Synopsis:* ex[it]
22919 x[it]

22920 Exit from **mailx** without changing the mailbox. No messages shall be saved in the **mbox** (see also
22921 **quit**).

22922 **Change Folder**

22923 *Synopsis:* fi[le] [file]
22924 fold[er] [file]

22925 Quit (see the **quit** command) from the current file of messages and read in the file named by the
22926 pathname *file*. If no argument is given, the name and status of the current mailbox shall be
22927 written.

22928 Several unquoted special characters shall be recognized when used as *file* names, with the
22929 following substitutions:

22930 % The system mailbox for the invoking user.
22931 %user The system mailbox for *user*.
22932 # The previous file.
22933 & The current **mbox**.
22934 +file The named file in the **folder** directory. (See the **folder** variable.)
22935 The default file shall be the current mailbox.

22936 **Display List of Folders**

22937 *Synopsis:* folders

22938 Write the names of the files in the directory set by the **folder** variable. The command specified by
22939 the *LISTER* environment variable shall be used (see the ENVIRONMENT VARIABLES section).

22940 **Follow Up Specified Messages**

22941 *Synopsis:* fo[llowup] [message]
22942 F[ollowup] [msglist]

22943 In the lowercase form, respond to a message, recording the response in a file whose name is
22944 derived from the author of the message. See also the **save** and **copy** commands and **outfolder**.

22945 In the capitalized form, respond to the first message in the *msglist*, sending the message to the
22946 author of each message in the *msglist*. The subject line shall be taken from the first message and
22947 the response shall be recorded in a file whose name is derived from the author of the first
22948 message. See also the **Save** and **Copy** commands and **outfolder**.

22949 Both forms shall override the **record** variable, if set.

22950 **Display Header Summary for Specified Messages**

22951 *Synopsis:* f[rom] [msglist]

22952 Write the header summary for the specified messages.

22953 **Display Header Summary**

22954 *Synopsis:* h[eaders] [message]

22955 Write the page of headers that includes the message specified. If the *message* argument is not
22956 specified, the current message shall not change. However, if the *message* argument is specified,
22957 the current message shall become the message that appears at the top of the page of headers that
22958 includes the message specified. The **screen** variable sets the number of headers per page. See
22959 also the **z** command.

22960 **Help**

22961 *Synopsis:* hel[p]
22962 ?

22963 Write a summary of commands.

22964 **Hold Messages**

22965 *Synopsis:* ho[ld] [msglist]
22966 pre[serve] [msglist]

22967 Mark the messages in *msglist* to be retained in the mailbox when *mailx* terminates. This shall
22968 override any commands that might previously have marked the messages to be deleted. During
22969 the current invocation of *mailx*, only the **delete**, **dp**, or **dt** commands shall remove the *preserve*
22970 marking of a message.

22971 **Execute Commands Conditionally**

22972 *Synopsis:* *i[f] s|r
mail-commands
el[se]
mail-commands
en[dif]*

22977 Execute commands conditionally, where **if** *s* executes the following *mail-commands*, up to an
22978 **else** or **endif**, if the program is in Send Mode, and **if** *r* shall cause the *mail-commands* to be
22979 executed only in Receive Mode.

22980 **List Available Commands**

22981 *Synopsis:* *l[ist]*

22982 Write a list of all commands available. No explanation shall be given.

22983 **Mail a Message**

22984 *Synopsis:* *mail address...*

22985 Mail a message to the specified addresses or aliases.

22986 **Direct Messages to mbox**

22987 *Synopsis:* *mb[ox] [msglist]*

22988 Arrange for the given messages to end up in the **mbox** save file when *mailx* terminates normally.
22989 See *MBOX*. See also the **exit** and **quit** commands.

22990 **Process Next Specified Message**

22991 *Synopsis:* *n[ext] [message]*

22992 If the current message has not been written (for example, by the **print** command) since *mailx*
22993 started or since any other message was the current message, behave as if the **print** command
22994 was entered. Otherwise, if there is an undeleted message after the current message, make it the
22995 current message and behave as if the **print** command was entered. Otherwise, an informational
22996 message to the effect that there are no further messages in the mailbox shall be written, followed
22997 by the *mailx* prompt.

22998 **Pipe Message**

22999 *Synopsis:* *pi[pe] [[msglist] command]
| [[msglist] command]*

23001 Pipe the messages through the given *command* by invoking the command interpreter specified
23002 by *SHELL* with two arguments: **-c** and *command*. (See also *sh -c*.) The application shall ensure
23003 that the command is given as a single argument. Quoting, described previously, can be used to
23004 accomplish this. If no arguments are given, the current message shall be piped through the
23005 command specified by the value of the **cmd** variable. If the **page** variable is set, a <form-feed>
23006 shall be inserted after each message.

23007 **Display Message with Headers**

23008 *Synopsis:* P[rint] [msglist]
23009 T[ype] [msglist]

23010 Write the specified messages, including all header lines, to standard output. Override
23011 suppression of lines by the **discard**, **ignore**, and **retain** commands. If **crt** is set, the messages
23012 longer than the number of lines specified by the **crt** variable shall be paged through the
23013 command specified by the *PAGER* environment variable.

23014 **Display Message**

23015 *Synopsis:* p[rint] [msglist]
23016 t[ype] [msglist]

23017 Write the specified messages to standard output. If **crt** is set, the messages longer than the
23018 number of lines specified by the **crt** variable shall be paged through the command specified by
23019 the *PAGER* environment variable.

23020 **Quit**

23021 *Synopsis:* q[uit]
23022 *end-of-file*

23023 Terminate *mailx*, storing messages that were read in **mbox** (if the current mailbox is the system
23024 mailbox and unless **hold** is set), deleting messages that have been explicitly saved (unless
23025 **keepsave** is set), discarding messages that have been deleted, and saving all remaining messages
23026 in the mailbox.

23027 **Reply to a Message List**

23028 *Synopsis:* R[eply] [msglist]
23029 R[espond] [msglist]

23030 Mail a reply message to the sender of each message in the *msglist*. The subject line shall be
23031 formed by concatenating **Re:<space>** (unless it already begins with that string) and the subject
23032 from the first message. If **record** is set to a filename, the response shall be saved at the end of that
23033 file.

23034 See also the **flipr** variable.

23035 **Reply to a Message**

23036 *Synopsis:* r[eply] [message]
23037 r[espond] [message]

23038 Mail a reply message to all recipients included in the header of the message. The subject line
23039 shall be formed by concatenating **Re:<space>** (unless it already begins with that string) and the
23040 subject from the message. If **record** is set to a filename, the response shall be saved at the end of
23041 that file.

23042 See also the **flipr** variable.

23043 **Retain Header Fields**

23044 *Synopsis:* `ret[ain] [header-field...]`

23045 Retain the specified header fields when writing messages. This command shall override all
23046 **discard** and **ignore** commands. The comparison of header fields shall be in a case-insensitive
23047 manner. If no arguments are specified, write a list of the currently retained header fields to
23048 standard output; the listing need not reflect the same order of header fields that were entered.

23049 **Save Messages**

23050 *Synopsis:* `s[ave] [file]`
23051 `s[ave] [msglist] file`
23052 `S[ave] [msglist]`

23053 Save the specified messages in the file named by the pathname *file*, or the **mbox** if the *file*
23054 argument is omitted. The file shall be created if it does not exist; otherwise, the messages shall be
23055 appended to the file. The message shall be put in the state *saved*, and shall behave as specified in
23056 the description of the *saved* state when the current mailbox is exited by the **quit** or **file**
23057 command.

23058 In the capitalized form, save the specified messages in a file whose name is derived from the
23059 author of the first message. The name of the file shall be taken to be the author's name with all
23060 network addressing stripped off. See also the **Copy**, **followup**, and **Followup** commands and
23061 **outfolder** variable.

23062 **Set Variables**

23063 *Synopsis:* `se[t] [name[=[string]] ...] [name=number ...] [noname ...]`

23064 Define one or more variables called *name*. The variable can be given a null, string, or numeric
23065 value. Quoting and backslash escapes can occur anywhere in *string*, as described previously, as
23066 if the *string* portion of the argument were the entire argument. The forms *name* and *name=* shall
23067 be equivalent to *name=""* for variables that take string values. The **set** command without
23068 arguments shall write a list of all defined variables and their values. The **no** *name* form shall be
23069 equivalent to **unset** *name*.

23070 **Invoke a Shell**

23071 *Synopsis:* `sh[ell]`

23072 Invoke an interactive command interpreter (see also **SHELL**).

23073 **Display Message Size**

23074 *Synopsis:* `si[ze] [msglist]`

23075 Write the size in bytes of each of the specified messages.

23076 **Read mailx Commands From a File**

23077 *Synopsis:* `so[urce] file`

23078 Read and execute commands from the file named by the pathname *file* and return to command
23079 mode.

23080 **Display Beginning of Messages**

23081 *Synopsis:* `to[p] [msglist]`

23082 Write the top few lines of each of the specified messages. If the **toplines** variable is set, it is taken
23083 as the number of lines to write. The default shall be 5.

23084 **Touch Messages**

23085 *Synopsis:* `tou[ch] [msglist]`

23086 Touch the specified messages. If any message in *msglist* is not specifically deleted nor saved in a
23087 file, it shall be placed in the **mbox** upon normal termination. See **exit** and **quit**.

23088 **Delete Aliases**

23089 *Synopsis:* `una[lias] [alias]...`

23090 Delete the specified alias names. If a specified alias does not exist, the results are unspecified.

23091 **Undelete Messages**

23092 *Synopsis:* `u[ndelete] [msglist]`

23093 Change the state of the specified messages from deleted to read. If **autoprint** is set, the last
23094 message of those restored shall be written. If *msglist* is not specified, the message shall be
23095 selected as follows:

- 23096 • If there are any deleted messages that follow the current message, the first of these shall be
23097 chosen.
- 23098 • Otherwise, the last deleted message that also precedes the current message shall be chosen.

23099 **Unset Variables**

23100 *Synopsis:* `uns[et] name...`

23101 Cause the specified variables to be erased.

23102 **Edit Message with Full-Screen Editor**

23103 *Synopsis:* `v[isual] [msglist]`

23104 Edit the given messages with a screen editor. Each message shall be placed in a temporary file,
23105 and the utility named by the **VISUAL** variable shall be invoked to edit each file in sequence. The
23106 default editor shall be *vi*.

23107 The **visual** command does not modify the contents of those messages in the mailbox.

23108 **Write Messages to a File**

23109 *Synopsis:* `w[rite] [msglist] file`

23110 Write the given messages to the file specified by the pathname *file*, minus the message header.
23111 Otherwise, it shall be equivalent to the **save** command.

23112 Scroll Header Display

23113 *Synopsis:* `z [+ | -]`

23114 Scroll the header display forward (if '+' is specified or if no option is specified) or backward (if
23115 '-' is specified) one screenful. The number of headers written shall be set by the **screen**
23116 variable.

23117 Invoke Shell Command

23118 *Synopsis:* `! command`

23119 Invoke the command interpreter specified by *SHELL* with two arguments: `-c` and *command*.
23120 (See also *sh -c*.) If the **bang** variable is set, each unescaped occurrence of '!' in *command* shall
23121 be replaced with the command executed by the previous ! command or "!" command escape.

23122 Null Command

23123 *Synopsis:* `# comment`

23124 This null command (comment) shall be ignored by *mailx*.

23125 Display Current Message Number

23126 *Synopsis:* `=`

23127 Write the current message number.

23128 Command Escapes in mailx

23129 The following commands can be entered only from input mode, by beginning a line with the
23130 escape character (by default, tilde ('~')). See the **escape** variable description for changing this
23131 special character. The format for the commands shall be:

23132 `<escape-character><command-char><separator>[<arguments>]`

23133 where the <separator> can be zero or more <blank>s.

23134 In the following descriptions, the application shall ensure that the argument *command* (but not
23135 *mailx-command*) is a shell command string. Any string acceptable to the command interpreter
23136 specified by the *SHELL* variable when it is invoked as *SHELL -c command_string* shall be valid.
23137 The command can be presented as multiple arguments (that is, quoting is not required).

23138 Command escapes that are listed with *msglist* or *mailx-command* arguments are invalid in Send
23139 Mode and produce unspecified results.

23140 `~! command` Invoke the command interpreter specified by *SHELL* with two arguments: `-c` and
23141 *command*; and then return to input mode. If the **bang** variable is set, each
23142 unescaped occurrence of '!' in *command* shall be replaced with the command
23143 executed by the previous ! command or "!" command escape.

23144 `~.` Simulate end-of-file (terminate message input).

23145 `~: mailx-command, ~_ mailx-command`
23146 Perform the command-level request.

23147 `~?` Write a summary of command escapes.

23148 `~A` This shall be equivalent to `~i Sign`.

23149 `~a` This shall be equivalent to `~i sign`.

- 23150 **~b** *name...* Add the *names* to the blind carbon copy (**Bcc**) list.
- 23151 **~c** *name...* Add the *names* to the carbon copy (**Cc**) list.
- 23152 **~d** Read in the dead-letter file. See *DEAD* for a description of this file.
- 23153 **~e** Invoke the editor, as specified by the *EDITOR* environment variable, on the partial message.
- 23155 **~f** [*msglist*] Forward the specified messages. The specified messages shall be inserted into the current message without alteration. This command escape also shall insert message headers into the message with field selection affected by the **discard**, **ignore**, and **retain** commands.
- 23156
- 23157
- 23158
- 23159 **~F** [*msglist*] This shall be the equivalent of the **~f** command escape, except that all headers shall be included in the message, regardless of previous **discard**, **ignore**, and **retain** commands.
- 23160
- 23161
- 23162 **~h** If standard input is a terminal, prompt for a **Subject** line and the **To**, **Cc**, and **Bcc** lists. Other implementation-defined headers may also be presented for editing. If the field is written with an initial value, it can be edited as if it had just been typed.
- 23163
- 23164
- 23165 **~i** *string* Insert the value of the named variable, followed by a <newline>, into the text of the message. If the string is unset or null, the message shall not be changed.
- 23166
- 23167 **~m** [*msglist*] Insert the specified messages into the message, prefixing non-empty lines with the string in the **indentprefix** variable. This command escape also shall insert message headers into the message, with field selection affected by the **discard**, **ignore**, and **retain** commands.
- 23168
- 23169
- 23170
- 23171 **~M** [*msglist*] This shall be the equivalent of the **~m** command escape, except that all headers shall be included in the message, regardless of previous **discard**, **ignore**, and **retain** commands.
- 23172
- 23173
- 23174 **~p** Write the message being entered. If the message is longer than **crt** lines (see **Internal Variables in mailx** (on page 590)), the output shall be paginated as described for the **PAGER** variable.
- 23175
- 23176
- 23177 **~q** Quit (see the **quit** command) from input mode by simulating an interrupt. If the body of the message is not empty, the partial message shall be saved in the dead-letter file. See *DEAD* for a description of this file.
- 23178
- 23179
- 23180 **~r** *file*, ~< *file*, ~r !*command*, ~< !*command*
- 23181 Read in the file specified by the pathname *file*. If the argument begins with an exclamation mark ('!'), the rest of the string shall be taken as an arbitrary system command; the command interpreter specified by *SHELL* shall be invoked with two arguments: **-c** and *command*. The standard output of *command* shall be inserted into the message.
- 23182
- 23183
- 23184
- 23185
- 23186 **~s** *string* Set the subject line to *string*.
- 23187 **~t** *name...* Add the given *names* to the **To** list.
- 23188 **~v** Invoke the full-screen editor, as specified by the *VISUAL* environment variable, on the partial message.
- 23189
- 23190 **~w** *file* Write the partial message, without the header, onto the file named by the pathname *file*. The file shall be created or the message shall be appended to it if the file exists.
- 23191
- 23192

23193 ~x Exit as with `~q`, except the message shall not be saved in the dead-letter file.
23194 ~| *command* Pipe the body of the message through the given *command* by invoking the
23195 command interpreter specified by *SHELL* with two arguments: `-c` and *command*.
23196 If the *command* returns a successful exit status, the standard output of the
23197 command shall replace the message. Otherwise, the message shall remain
23198 unchanged. If the *command* fails, an error message giving the exit status shall be
23199 written.

23200 EXIT STATUS

23201 When the `-e` option is specified, the following exit values are returned:

23202 0 Mail was found.

23203 >0 Mail was not found or an error occurred.

23204 Otherwise, the following exit values are returned:

23205 0 Successful completion; note that this status implies that all messages were *sent*, but it gives
23206 no assurances that any of them were actually *delivered*.

23207 >0 An error occurred.

23208 CONSEQUENCES OF ERRORS

23209 When in input mode (Receive Mode) or Send Mode:

- If an error is encountered processing a command escape (see **Command Escapes in mailx** (on page 601)), a diagnostic message shall be written to standard error, and the message being composed may be modified, but this condition shall not prevent the message from being sent.
- Other errors shall prevent the sending of the message.

23215 When in command mode:

- Default.

23217 APPLICATION USAGE

23218 Delivery of messages to remote systems requires the existence of communication paths to such
23219 systems. These need not exist.

23220 Input lines are limited to {LINE_MAX} bytes, but mailers between systems may impose more
23221 severe line-length restrictions. This volume of IEEE Std 1003.1-2001 does not place any
23222 restrictions on the length of messages handled by *mailx*, and for delivery of local messages the
23223 only limitations should be the normal problems of available disk space for the target mail file.
23224 When sending messages to external machines, applications are advised to limit messages to less
23225 than 100 000 bytes because some mail gateways impose message-length restrictions.

23226 The format of the system mailbox is intentionally unspecified. Not all systems implement
23227 system mailboxes as flat files, particularly with the advent of multimedia mail messages. Some
23228 system mailboxes may be multiple files, others records in a database. The internal format of the
23229 messages themselves is specified with the historical format from Version 7, but only after the
23230 messages have been saved in some file other than the system mailbox. This was done so that
23231 many historical applications expecting text-file mailboxes are not broken.

23232 Some new formats for messages can be expected in the future, probably including binary data,
23233 bit maps, and various multimedia objects. As described here, *mailx* is not prohibited from
23234 handling such messages, but it must store them as text files in secondary mailboxes (unless
23235 some extension, such as a variable or command line option, is used to change the stored format).
23236 Its method of doing so is implementation-defined and might include translating the data into

23237 text file-compatible or readable form or omitting certain portions of the message from the stored
23238 output.

23239 The **discard** and **ignore** commands are not inverses of the **retain** command. The **retain**
23240 command discards all header-fields except those explicitly retained. The **discard** command
23241 keeps all header-fields except those explicitly discarded. If headers exist on the retained header
23242 list, **discard** and **ignore** commands are ignored.

23243 **EXAMPLES**

23244 None.

23245 **RATIONALE**

23246 The standard developers felt strongly that a method for applications to send messages to
23247 specific users was necessary. The obvious example is a batch utility, running non-interactively,
23248 that wishes to communicate errors or results to a user. However, the actual format, delivery
23249 mechanism, and method of reading the message are clearly beyond the scope of this volume of
23250 IEEE Std 1003.1-2001.

23251 The intent of this command is to provide a simple, portable interface for sending messages non-
23252 interactively. It merely defines a “front-end” to the historical mail system. It is suggested that
23253 implementations explicitly denote the sender and recipient in the body of the delivered message.
23254 Further specification of formats for either the message envelope or the message itself were
23255 deliberately not made, as the industry is in the midst of changing from the current standards to a
23256 more internationalized standard and it is probably incorrect, at this time, to require either one.

23257 Implementations are encouraged to conform to the various delivery mechanisms described in
23258 the CCITT X.400 standards or to the equivalent Internet standards, described in Internet Request
23259 for Comment (RFC) documents RFC 819, RFC 822, RFC 920, RFC 921, and RFC 1123.

23260 Many historical systems modified each body line that started with **From** by prefixing the ‘F’
23261 with ‘>’. It is unnecessary, but allowed, to do that when the string does not follow a blank line
23262 because it cannot be confused with the next header.

23263 The **edit** and **visual** commands merely edit the specified messages in a temporary file. They do
23264 not modify the contents of those messages in the mailbox; such a capability could be added as an
23265 extension, such as by using different command names.

23266 The restriction on a subject line being {LINE_MAX}-10 bytes is based on the historical format
23267 that consumes 10 bytes for **Subject:** and the trailing <newline>. Many historical mailers that a
23268 message may encounter on other systems are not able to handle lines that long, however.

23269 Like the utilities *logger* and *lp*, *mailx* admittedly is difficult to test. This was not deemed sufficient
23270 justification to exclude this utility from this volume of IEEE Std 1003.1-2001. It is also arguable
23271 that it is, in fact, testable, but that the tests themselves are not portable.

23272 When *mailx* is being used by an application that wishes to receive the results as if none of the
23273 User Portability Utilities option features were supported, the **DEAD** environment variable must
23274 be set to **/dev/null**. Otherwise, it may be subject to the file creations described in *mailx*
23275 ASYNCHRONOUS EVENTS. Similarly, if the **MAILRC** environment variable is not set to
23276 **/dev/null**, historical versions of *mailx* and *Mail* read initialization commands from a file before
23277 processing begins. Since the initialization that a user specifies could alter the contents of
23278 messages an application is trying to send, such applications must set **MAILRC** to **/dev/null**.

23279 The description of **LC_TIME** uses “may affect” because many historical implementations do not
23280 or cannot manipulate the date and time strings in the incoming mail headers. Some headers
23281 found in incoming mail do not have enough information to determine the timezone in which the
23282 mail originated, and, therefore, *mailx* cannot convert the date and time strings into the internal
23283 form that then is parsed by routines like *strftime()* that can take **LC_TIME** settings into account.

23284 Changing all these times to a user-specified format is allowed, but not required.

23285 The paginator selected when *PAGER* is null or unset is partially unspecified to allow the System
23286 V historical practice of using *pg* as the default. Bypassing the pagination function, such as by
23287 declaring that *cat* is the paginator, would not meet with the intended meaning of this
23288 description. However, any “portable user” would have to set *PAGER* explicitly to get his or her
23289 preferred paginator on all systems. The paginator choice was made partially unspecified, unlike
23290 the *VISUAL* editor choice (mandated to be *vi*) because most historical pagers follow a common
23291 theme of user input, whereas editors differ dramatically.

23292 Options to specify addresses as **cc** (carbon copy) or **bcc** (blind carbon copy) were considered to
23293 be format details and were omitted.

23294 A zero exit status implies that all messages were *sent*, but it gives no assurances that any of them
23295 were actually *delivered*. The reliability of the delivery mechanism is unspecified and is an
23296 appropriate marketing distinction between systems.

23297 In order to conform to the Utility Syntax Guidelines, a solution was required to the optional *file*
23298 option-argument to **-f**. By making *file* an operand, the guidelines are satisfied and users remain
23299 portable. However, it does force implementations to support usage such as:

23300 mailx -fin mymail.box

23301 The **no name** method of unsetting variables is not present in all historical systems, but it is in
23302 System V and provides a logical set of commands corresponding to the format of the display of
23303 options from the *mailx set* command without arguments.

23304 The **ask** and **asksub** variables are the names selected by BSD and System V, respectively, for the
23305 same feature. They are synonyms in this volume of IEEE Std 1003.1-2001.

23306 The *mailx echo* command was not documented in the BSD version and has been omitted here
23307 because it is not obviously useful for interactive users.

23308 The default prompt on the System V *mailx* is a question mark, on BSD *Mail* an ampersand. Since
23309 this volume of IEEE Std 1003.1-2001 chose the *mailx* name, it kept the System V default,
23310 assuming that BSD users would not have difficulty with this minor incompatibility (that they
23311 can override).

23312 The meanings of **r** and **R** are reversed between System V *mailx* and SunOS *Mail*. Once again,
23313 since this volume of IEEE Std 1003.1-2001 chose the *mailx* name, it kept the System V default, but
23314 allows the SunOS user to achieve the desired results using **flipr**, an internal variable in System V
23315 *mailx*, although it has not been documented in the SVID.

23316 The **indentprefix** variable, the **retain** and **unalias** commands, and the **~F** and **~M** command
23317 escapes were adopted from 4.3 BSD *Mail*.

23318 The **version** command was not included because no sufficiently general specification of the
23319 version information could be devised that would still be useful to a portable user. This
23320 command name should be used by suppliers who wish to provide version information about the
23321 *mailx* command.

23322 The “implementation-specific (unspecified) system start-up file” historically has been named
23323 */etc/mailx.rc*, but this specific name and location are not required.

23324 The intent of the wording for the **next** command is that if any command has already displayed
23325 the current message it should display a following message, but, otherwise, it should display the
23326 current message. Consider the command sequence:

23327 next 3
23328 delete 3

23329 next
23330 where the **autoprint** option was not set. The normative text specifies that the second **next**
23331 command should display a message following the third message, because even though the
23332 current message has not been displayed since it was set by the **delete** command, it has been
23333 displayed since the current message was anything other than message number 3. This does not
23334 always match historical practice in some implementations, where the command file address
23335 followed by **next** (or the default command) would skip the message for which the user had
23336 searched.

23337 FUTURE DIRECTIONS

23338 None.

23339 SEE ALSO

23340 Chapter 2 (on page 29), *ed*, *ls*, *more*, *vi*

23341 CHANGE HISTORY

23342 First released in Issue 2.

23343 Issue 5

23344 The description of the *EDITOR* environment variable is changed to indicate that *ed* is the default
23345 editor if this variable is not set. In previous issues, this default was not stated explicitly at this
23346 point but was implied further down in the text.

23347 The FUTURE DIRECTIONS section is added.

23348 Issue 6

23349 The following new requirements on POSIX implementations derive from alignment with the
23350 Single UNIX Specification:

- 23351 • The **-F** option is added.
- 23352 • The **allnet**, **debug**, and **sendwait** internal variables are added.
- 23353 • The **C**, **ec**, **fo**, **F**, and **S** *mailx* commands are added.

23354 In the DESCRIPTION and ENVIRONMENT VARIABLES sections, text stating “*HOME*
23355 directory” is replaced by “directory referred to by the *HOME* environment variable”.

23356 The *mailx* utility is aligned with the IEEE P1003.2b draft standard, which includes various
23357 clarifications to resolve IEEE PASC Interpretations submitted for the ISO POSIX-2:1993
23358 standard. In particular, the changes here address IEEE PASC Interpretations 1003.2 #10, #11,
23359 #103, #106, #108, #114, #115, #122, and #129.

23360 The normative text is reworded to avoid use of the term “must” for application requirements.

23361 The **TZ** entry is added to the ENVIRONMENT VARIABLES section.

23362 NAME

23363 make — maintain, update, and regenerate groups of programs (DEVELOPMENT)

23364 SYNOPSIS

23365 SD make [-einpqrst][-f *makefile*]...[-k | -S][macro=value]...
23366 [target_name...]
23367

23368 DESCRIPTION

23369 The *make* utility shall update files that are derived from other files. A typical case is one where
23370 object files are derived from the corresponding source files. The *make* utility examines time
23371 relationships and shall update those derived files (called targets) that have modified times
23372 earlier than the modified times of the files (called prerequisites) from which they are derived. A
23373 description file (makefile) contains a description of the relationships between files, and the
23374 commands that need to be executed to update the targets to reflect changes in their
23375 prerequisites. Each specification, or rule, shall consist of a target, optional prerequisites, and
23376 optional commands to be executed when a prerequisite is newer than the target. There are two
23377 types of rule:

- 23378 1. *Inference rules*, which have one target name with at least one period ('.') and no slash
23379 ('/')

- 23380 2. *Target rules*, which can have more than one target name

23381 In addition, *make* shall have a collection of built-in macros and inference rules that infer
23382 prerequisite relationships to simplify maintenance of programs.

23383 To receive exactly the behavior described in this section, the user shall ensure that a portable
23384 makefile shall:

- 23385 • Include the special target .POSIX
- 23386 • Omit any special target reserved for implementations (a leading period followed by
23387 uppercase letters) that has not been specified by this section

23388 The behavior of *make* is unspecified if either or both of these conditions are not met.

23389 OPTIONS

23390 The *make* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
23391 12.2, Utility Syntax Guidelines.

23392 The following options shall be supported:

- 23393 **-e** Cause environment variables, including those with null values, to override macro
23394 assignments within makefiles.
- 23395 **-f *makefile*** Specify a different makefile. The argument *makefile* is a pathname of a description
23396 file, which is also referred to as the *makefile*. A pathname of '-' shall denote the
23397 standard input. There can be multiple instances of this option, and they shall be
23398 processed in the order specified. The effect of specifying the same option-
23399 argument more than once is unspecified.
- 23400 **-i** Ignore error codes returned by invoked commands. This mode is the same as if the
23401 special target .IGNORE were specified without prerequisites.
- 23402 **-k** Continue to update other targets that do not depend on the current target if a non-
23403 ignored error occurs while executing the commands to bring a target up-to-date.
- 23404 **-n** Write commands that would be executed on standard output, but do not execute
23405 them. However, lines with a plus sign ('+') prefix shall be executed. In this mode,

23406 lines with an at sign ('@') character prefix shall be written to standard output.

23407 **-p** Write to standard output the complete set of macro definitions and target descriptions. The output format is unspecified.

23409 **-q** Return a zero exit value if the target file is up-to-date; otherwise, return an exit value of 1. Targets shall not be updated if this option is specified. However, a makefile command line (associated with the targets) with a plus sign ('+') prefix shall be executed.

23413 **-r** Clear the suffix list and do not use the built-in rules.

23414 **-S** Terminate *make* if an error occurs while executing the commands to bring a target up-to-date. This shall be the default and the opposite of **-k**.

23416 **-s** Do not write makefile command lines or touch messages (see **-t**) to standard output before executing. This mode shall be the same as if the special target **.SILENT** were specified without prerequisites.

23419 **-t** Update the modification time of each target as though a *touch target* had been executed. Targets that have prerequisites but no commands (see **Target Rules** (on page 611)), or that are already up-to-date, shall not be touched in this manner. Write messages to standard output for each target file indicating the name of the file and that it was touched. Normally, the *makefile* command lines associated with each target are not executed. However, a command line with a plus sign ('+') prefix shall be executed.

23426 Any options specified in the *MAKEFLAGS* environment variable shall be evaluated before any options specified on the *make* utility command line. If the **-k** and **-S** options are both specified on the *make* utility command line or by the *MAKEFLAGS* environment variable, the last option specified shall take precedence. If the **-f** or **-p** options appear in the *MAKEFLAGS* environment variable, the result is undefined.

23431 OPERANDS

23432 The following operands shall be supported:

23433 *target_name* Target names, as defined in the EXTENDED DESCRIPTION section. If no target is specified, while *make* is processing the makefiles, the first target that *make* encounters that is not a special target or an inference rule shall be used.

23436 *macro=value* Macro definitions, as defined in **Macros** (on page 613).

23437 If the *target_name* and *macro=value* operands are intermixed on the *make* utility command line, the results are unspecified.

23439 STDIN

23440 The standard input shall be used only if the *makefile* option-argument is '**-**'. See the INPUT FILES section.

23442 INPUT FILES

23443 The input file, otherwise known as the makefile, is a text file containing rules, macro definitions, and comments. See the EXTENDED DESCRIPTION section.

23445 ENVIRONMENT VARIABLES

23446 The following environment variables shall affect the execution of *make*:

23447 **LANG** Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

23451	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
23452		
23453	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).
23454		
23455		
23456	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
23457		
23458		
23459	<i>MAKEFLAGS</i>	
23460		This variable shall be interpreted as a character string representing a series of option characters to be used as the default options. The implementation shall accept both of the following formats (but need not accept them when intermixed):
23461		
23462		
23463		<ul style="list-style-type: none"> • The characters are option letters without the leading hyphens or <blank> separation used on a <i>make</i> utility command line.
23464		
23465		<ul style="list-style-type: none"> • The characters are formatted in a manner similar to a portion of the <i>make</i> utility command line: options are preceded by hyphens and <blank>-separated as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines. The <i>macro=value</i> macro definition operands can also be included. The difference between the contents of <i>MAKEFLAGS</i> and the <i>make</i> utility command line is that the contents of the variable shall not be subjected to the word expansions (see Section 2.6 (on page 36)) associated with parsing the command line values.
23466		
23467		
23468		
23469		
23470		
23471		
23472		
23473 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
23474 XSI	<i>PROJECTDIR</i>	<p>Provide a directory to be used to search for SCCS files not found in the current directory. In all of the following cases, the search for SCCS files is made in the directory SCCS in the identified directory. If the value of <i>PROJECTDIR</i> begins with a slash, it shall be considered an absolute pathname; otherwise, the value of <i>PROJECTDIR</i> is treated as a user name and that user's initial working directory shall be examined for a subdirectory src or source. If such a directory is found, it shall be used. Otherwise, the value is used as a relative pathname.</p> <p>If <i>PROJECTDIR</i> is not set or has a null value, the search for SCCS files shall be made in the directory SCCS in the current directory.</p> <p>The setting of <i>PROJECTDIR</i> affects all files listed in the remainder of this utility description for files with a component named SCCS.</p> <p>The value of the SHELL environment variable shall not be used as a macro and shall not be modified by defining the SHELL macro in a makefile or on the command line. All other environment variables, including those with null values, shall be used as macros, as defined in Macros (on page 613).</p>
23475		
23476		
23477		
23478		
23479		
23480		
23481		
23482		
23483		
23484		
23485		
23486		
23487		
23488		
23489		
23490	ASYNCHRONOUS EVENTS	
23491		If not already ignored, <i>make</i> shall trap SIGHUP, SIGTERM, SIGINT, and SIGQUIT and remove the current target unless the target is a directory or the target is a prerequisite of the special target .PRECIOUS or unless one of the -n , -p , or -q options was specified. Any targets removed in this manner shall be reported in diagnostic messages of unspecified format, written to standard error. After this cleanup process, if any, <i>make</i> shall take the standard action for all other signals.
23492		
23493		
23494		
23495		
23496		

23497 **STDOUT**

23498 The *make* utility shall write all commands to be executed to standard output unless the **-s** option
23499 was specified, the command is prefixed with an at sign, or the special target **.SILENT** has either
23500 the current target as a prerequisite or has no prerequisites. If *make* is invoked without any work
23501 needing to be done, it shall write a message to standard output indicating that no action was
23502 taken. If the **-t** option is present and a file is touched, *make* shall write to standard output a
23503 message of unspecified format indicating that the file was touched, including the filename of the
23504 file.

23505 **STDERR**

23506 The standard error shall be used only for diagnostic messages.

23507 **OUTPUT FILES**

23508 Files can be created when the **-t** option is present. Additional files can also be created by the
23509 utilities invoked by *make*.

23510 **EXTENDED DESCRIPTION**

23511 The *make* utility attempts to perform the actions required to ensure that the specified targets are
23512 up-to-date. A target is considered out-of-date if it is older than any of its prerequisites or if it
23513 does not exist. The *make* utility shall treat all prerequisites as targets themselves and recursively
23514 ensure that they are up-to-date, processing them in the order in which they appear in the rule.
23515 The *make* utility shall use the modification times of files to determine whether the corresponding
23516 targets are out-of-date.

23517 After *make* has ensured that all of the prerequisites of a target are up-to-date and if the target is
23518 out-of-date, the commands associated with the target entry shall be executed. If there are no
23519 commands listed for the target, the target shall be treated as up-to-date.

23520 **Makefile Syntax**

23521 A makefile can contain rules, macro definitions (see **Macros** (on page 613)), and comments.
23522 There are two kinds of rules: *inference rules* and *target rules*. The *make* utility shall contain a set of
23523 built-in inference rules. If the **-r** option is present, the built-in rules shall not be used and the
23524 suffix list shall be cleared. Additional rules of both types can be specified in a makefile. If a rule
23525 is defined more than once, the value of the rule shall be that of the last one specified. Macros can
23526 also be defined more than once, and the value of the macro is specified in **Macros** (on page 613).
23527 Comments start with a number sign ('#') and continue until an unescaped <newline> is
23528 reached.

23529 By default, the following files shall be tried in sequence: **./makefile** and **./Makefile**. If neither
23530 **./makefile** or **./Makefile** are found, other implementation-defined files may also be tried. On
23531 XSI-conformant systems, the additional files **./s.makefile**, **SCCS/s.makefile**, **./s.Makefile**, and
23532 **SCCS/s.Makefile** shall also be tried.

23533 The **-f** option shall direct *make* to ignore any of these default files and use the specified argument
23534 as a makefile instead. If the '-' argument is specified, standard input shall be used.

23535 The term *makefile* is used to refer to any rules provided by the user, whether in **./makefile** or its
23536 variants, or specified by the **-f** option.

23537 The rules in makefiles shall consist of the following types of lines: target rules, including special
23538 targets (see **Target Rules** (on page 611)), inference rules (see **Inference Rules** (on page 614)),
23539 macro definitions (see **Macros** (on page 613)), empty lines, and comments.

23540 When an escaped <newline> (one preceded by a backslash) is found anywhere in the makefile
23541 except in a command line, it shall be replaced, along with any leading white space on the
23542 following line, with a single <space>. When an escaped <newline> is found in a command line

23543 in a makefile, the command line shall contain the backslash, the <newline>, and the next line,
23544 except that the first character of the next line shall not be included if it is a <tab>.

23545 Makefile Execution

23546 Makefile command lines shall be processed one at a time by writing the makefile command line
23547 to the standard output (unless one of the conditions listed under '@' suppresses the writing)
23548 and executing the command(s) in the line. A <tab> may precede the command to standard
23549 output. Command execution shall be as if the makefile command line were the argument to the
23550 *system()* function. The environment for the command being executed shall contain all of the
23551 variables in the environment of *make*.

23552 By default, when *make* receives a non-zero status from the execution of a command, it shall
23553 terminate with an error message to standard error.

23554 Makefile command lines can have one or more of the following prefixes: a hyphen ('-'), an at
23555 sign ('@'), or a plus sign ('+'). These shall modify the way in which *make* processes the
23556 command. When a command is written to standard output, the prefix shall not be included in
23557 the output.

- 23558 – If the command prefix contains a hyphen, or the **-i** option is present, or the special target
23559 **.IGNORE** has either the current target as a prerequisite or has no prerequisites, any error
23560 found while executing the command shall be ignored.
- 23561 @ If the command prefix contains an at sign and the *make* utility command line **-n** option is
23562 not specified, or the **-s** option is present, or the special target **.SILENT** has either the current
23563 target as a prerequisite or has no prerequisites, the command shall not be written to
23564 standard output before it is executed.
- 23565 + If the command prefix contains a plus sign, this indicates a makefile command line that
23566 shall be executed even if **-n**, **-q**, or **-t** is specified.

23567 Target Rules

23568 Target rules are formatted as follows:

```
23569 target [target...]: [prerequisite...] [;command]  
23570 [<tab>command  
23571 <tab>command  
23572 ...]  
23573 line that does not begin with <tab>
```

23574 Target entries are specified by a <blank>-separated, non-null list of targets, then a colon, then a
23575 <blank>-separated, possibly empty list of prerequisites. Text following a semicolon, if any, and
23576 all following lines that begin with a <tab>, are makefile command lines to be executed to update
23577 the target. The first non-empty line that does not begin with a <tab> or '#' shall begin a new
23578 entry. An empty or blank line, or a line beginning with '#', may begin a new entry.

23579 Applications shall select target names from the set of characters consisting solely of periods,
23580 underscores, digits, and alphabets from the portable character set (see the Base Definitions
23581 volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set). Implementations may allow
23582 other characters in target names as extensions. The interpretation of targets containing the
23583 characters '%' and '"' is implementation-defined.

23584 A target that has prerequisites, but does not have any commands, can be used to add to the
23585 prerequisite list for that target. Only one target rule for any given target can contain commands.

23586	Lines that begin with one of the following are called <i>special targets</i> and control the operation of
23587	make:
23588	.DEFAULT If the makefile uses this special target, the application shall ensure that it is
23589	specified with commands, but without prerequisites. The commands shall be used
23590	by <i>make</i> if there are no other rules available to build a target.
23591	.IGNORE Prerequisites of this special target are targets themselves; this shall cause errors
23592	from commands associated with them to be ignored in the same manner as
23593	specified by the -i option. Subsequent occurrences of .IGNORE shall add to the
23594	list of targets ignoring command errors. If no prerequisites are specified, <i>make</i> shall
23595	behave as if the -i option had been specified and errors from all commands
23596	associated with all targets shall be ignored.
23597	.POSIX The application shall ensure that this special target is specified without
23598	prerequisites or commands. If it appears as the first non-comment line in the
23599	makefile, <i>make</i> shall process the makefile as specified by this section; otherwise, the
23600	behavior of <i>make</i> is unspecified.
23601	.PRECIOUS Prerequisites of this special target shall not be removed if <i>make</i> receives one of the
23602	asynchronous events explicitly described in the ASYNCHRONOUS EVENTS
23603	section. Subsequent occurrences of .PRECIOUS shall add to the list of precious
23604	files. If no prerequisites are specified, all targets in the makefile shall be treated as
23605	if specified with .PRECIOUS .
23606 XSI	.SCCS_GET The application shall ensure that this special target is specified without
23607	prerequisites. If this special target is included in a makefile, the commands
23608	specified with this target shall replace the default commands associated with this
23609	special target (see Default Rules (on page 617)). The commands specified with
23610	this target are used to get all SCCS files that are not found in the current directory.
23611	When source files are named in a dependency list, <i>make</i> shall treat them just like
23612	any other target. Because the source file is presumed to be present in the directory,
23613	there is no need to add an entry for it to the makefile. When a target has no
23614	dependencies, but is present in the directory, <i>make</i> shall assume that that file is up-
23615	to-date. If, however, an SCCS file named SCCS/s.source_file is found for a target
23616	<i>source_file</i> , <i>make</i> compares the timestamp of the target file with that of the
23617	SCCS/s.source_file to ensure the target is up-to-date. If the target is missing, or if
23618	the SCCS file is newer, <i>make</i> shall automatically issue the commands specified for
23619	the .SCCS_GET special target to retrieve the most recent version. However, if the
23620	target is writable by anyone, <i>make</i> shall not retrieve a new version.
23621	.SILENT Prerequisites of this special target are targets themselves; this shall cause
23622	commands associated with them not to be written to the standard output before
23623	they are executed. Subsequent occurrences of .SILENT shall add to the list of
23624	targets with silent commands. If no prerequisites are specified, <i>make</i> shall behave
23625	as if the -s option had been specified and no commands or touch messages
23626	associated with any target shall be written to standard output.
23627	.SUFFIXES Prerequisites of .SUFFIXES shall be appended to the list of known suffixes and are
23628	used in conjunction with the inference rules (see Inference Rules (on page 614)). If
23629	.SUFFIXES does not have any prerequisites, the list of known suffixes shall be
23630	cleared.
23631	The special targets .IGNORE , .POSIX , .PRECIOUS , .SILENT , and .SUFFIXES shall be specified
23632	without commands.

23633 Targets with names consisting of a leading period followed by the uppercase letters "POSIX"
23634 and then any other characters are reserved for future standardization. Targets with names
23635 consisting of a leading period followed by one or more uppercase letters are reserved for
23636 implementation extensions.

23637 **Macros**

23638 Macro definitions are in the form:

23639 *string1* = [*string2*]

23640 The macro named *string1* is defined as having the value of *string2*, where *string2* is defined as all
23641 characters, if any, after the equal sign, up to a comment character ('#'') or an unescaped
23642 <newline>. Any <blank>s immediately before or after the equal sign shall be ignored.

23643 Applications shall select macro names from the set of characters consisting solely of periods,
23644 underscores, digits, and alphabets from the portable character set (see the Base Definitions
23645 volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set). A macro name shall not
23646 contain an equals sign. Implementations may allow other characters in macro names as
23647 extensions.

23648 Macros can appear anywhere in the makefile. Macro expansions using the forms `$(string1)` or
23649 `${string1}` shall be replaced by *string2*, as follows:

- 23650 • Macros in target lines shall be evaluated when the target line is read.
23651 • Macros in makefile command lines shall be evaluated when the command is executed.
23652 • Macros in the string before the equals sign in a macro definition shall be evaluated when the
23653 macro assignment is made.
23654 • Macros after the equals sign in a macro definition shall not be evaluated until the defined
23655 macro is used in a rule or command, or before the equals sign in a macro definition.

23656 The parentheses or braces are optional if *string1* is a single character. The macro `$$` shall be
23657 replaced by the single character '\$'. If *string1* in a macro expansion contains a macro
23658 expansion, the results are unspecified.

23659 Macro expansions using the forms `$(string1[:subst1=subst2])` or `${string1[:subst1=subst2]}` can
23660 be used to replace all occurrences of *subst1* with *subst2* when the macro substitution is
23661 performed. The *subst1* to be replaced shall be recognized when it is a suffix at the end of a word
23662 in *string1* (where a *word*, in this context, is defined to be a string delimited by the beginning of
23663 the line, a <blank>, or a <newline>). If *string1* in a macro expansion contains a macro expansion,
23664 the results are unspecified.

23665 Macro expansions in *string1* of macro definition lines shall be evaluated when read. Macro
23666 expansions in *string2* of macro definition lines shall be performed when the macro identified by
23667 *string1* is expanded in a rule or command.

23668 Macro definitions shall be taken from the following sources, in the following logical order,
23669 before the makefile(s) are read.

- 23670 1. Macros specified on the *make* utility command line, in the order specified on the command
23671 line. It is unspecified whether the internal macros defined in **Internal Macros** (on page 616)
23672 are accepted from this source.
- 23673 2. Macros defined by the *MAKEFLAGS* environment variable, in the order specified in the
23674 environment variable. It is unspecified whether the internal macros defined in **Internal
23675 Macros** (on page 616) are accepted from this source.

23676 3. The contents of the environment, excluding the *MAKEFLAGS* and *SHELL* variables and
23677 including the variables with null values.

23678 4. Macros defined in the inference rules built into *make*.

23679 Macro definitions from these sources shall not override macro definitions from a lower-
23680 numbered source. Macro definitions from a single source (for example, the *make* utility
23681 command line, the *MAKEFLAGS* environment variable, or the other environment variables) shall
23682 override previous macro definitions from the same source.

23683 Macros defined in the makefile(s) shall override macro definitions that occur before them in the
23684 makefile(s) and macro definitions from source 4. If the **-e** option is not specified, macros defined
23685 in the makefile(s) shall override macro definitions from source 3. Macros defined in the
23686 makefile(s) shall not override macro definitions from source 1 or source 2.

23687 Before the makefile(s) are read, all of the *make* utility command line options (except **-f** and **-p**)
23688 and *make* utility command line macro definitions (except any for the *MAKEFLAGS* macro), not
23689 already included in the *MAKEFLAGS* macro, shall be added to the *MAKEFLAGS* macro, quoted
23690 in an implementation-defined manner such that when *MAKEFLAGS* is read by another instance
23691 of the *make* command, the original macro's value is recovered. Other implementation-defined
23692 options and macros may also be added to the *MAKEFLAGS* macro. If this modifies the value of
23693 the *MAKEFLAGS* macro, or, if the *MAKEFLAGS* macro is modified at any subsequent time, the
23694 *MAKEFLAGS* environment variable shall be modified to match the new value of the
23695 *MAKEFLAGS* macro. The result of setting *MAKEFLAGS* in the Makefile is unspecified.

23696 Before the makefile(s) are read, all of the *make* utility command line macro definitions (except the
23697 *MAKEFLAGS* macro or the *SHELL* macro) shall be added to the environment of *make*. Other
23698 implementation-defined variables may also be added to the environment of *make*.

23699 The **SHELL** macro shall be treated specially. It shall be provided by *make* and set to the
23700 pathname of the shell command language interpreter (see *sh*). The **SHELL** environment variable
23701 shall not affect the value of the **SHELL** macro. If **SHELL** is defined in the makefile or is specified
23702 on the command line, it shall replace the original value of the **SHELL** macro, but shall not affect
23703 the **SHELL** environment variable. Other effects of defining **SHELL** in the makefile or on the
23704 command line are implementation-defined.

23705 Inference Rules

23706 Inference rules are formatted as follows:

```
23707        target:  
23708        <tab>command  
23709        [<tab>command]  
23710        ...  
23711        line that does not begin with <tab> or #
```

23712 The application shall ensure that the *target* portion is a valid target name (see **Target Rules** (on
23713 page 611)) of the form *.s2* or *.s1.s2* (where *.s1* and *.s2* are suffixes that have been given as
23714 prerequisites of the **.SUFFIXES** special target and *s1* and *s2* do not contain any slashes or
23715 periods.) If there is only one period in the target, it is a single-suffix inference rule. Targets with
23716 two periods are double-suffix inference rules. Inference rules can have only one target before the
23717 colon.

23718 The application shall ensure that the makefile does not specify prerequisites for inference rules;
23719 no characters other than white space shall follow the colon in the first line, except when creating
23720 the *empty rule*, described below. Prerequisites are inferred, as described below.

23721 Inference rules can be redefined. A target that matches an existing inference rule shall overwrite
23722 the old inference rule. An empty rule can be created with a command consisting of simply a
23723 semicolon (that is, the rule still exists and is found during inference rule search, but since it is
23724 empty, execution has no effect). The empty rule can also be formatted as follows:

23725 `rule: ;`

23726 where zero or more <blank>s separate the colon and semicolon.

23727 The *make* utility uses the suffixes of targets and their prerequisites to infer how a target can be
23728 made up-to-date. A list of inference rules defines the commands to be executed. By default, *make*
23729 contains a built-in set of inference rules. Additional rules can be specified in the makefile.

23730 The special target **.SUFFIXES** contains as its prerequisites a list of suffixes that shall be used by
23731 the inference rules. The order in which the suffixes are specified defines the order in which the
23732 inference rules for the suffixes are used. New suffixes shall be appended to the current list by
23733 specifying a **.SUFFIXES** special target in the makefile. A **.SUFFIXES** target with no prerequisites
23734 shall clear the list of suffixes. An empty **.SUFFIXES** target followed by a new **.SUFFIXES** list is
23735 required to change the order of the suffixes.

23736 Normally, the user would provide an inference rule for each suffix. The inference rule to update
23737 a target with a suffix **.s1** from a prerequisite with a suffix **.s2** is specified as a target **.s2.s1**. The
23738 internal macros provide the means to specify general inference rules (see **Internal Macros** (on
23739 page 616)).

23740 When no target rule is found to update a target, the inference rules shall be checked. The suffix
23741 of the target (**.s1**) to be built is compared to the list of suffixes specified by the **.SUFFIXES** special
23742 targets. If the **.s1** suffix is found in **.SUFFIXES**, the inference rules shall be searched in the order
23743 defined for the first **.s2.s1** rule whose prerequisite file (**\$*.s2**) exists. If the target is out-of-date
23744 with respect to this prerequisite, the commands for that inference rule shall be executed.

23745 If the target to be built does not contain a suffix and there is no rule for the target, the single
23746 suffix inference rules shall be checked. The single-suffix inference rules define how to build a
23747 target if a file is found with a name that matches the target name with one of the single suffixes
23748 appended. A rule with one suffix **.s2** is the definition of how to build *target* from **target.s2**. The
23749 other suffix (**.s1**) is treated as null.

23750 XSI A tilde ('~') in the above rules refers to an SCCS file in the current directory. Thus, the rule **.c~.o**
23751 would transform an SCCS C-language source file into an object file (.o). Because the **s.** of the
23752 SCCS files is a prefix, it is incompatible with *make*'s suffix point of view. Hence, the '~' is a way
23753 of changing any file reference into an SCCS file reference.

23754 Libraries

23755 If a target or prerequisite contains parentheses, it shall be treated as a member of an archive
23756 library. For the **lib(member.o)** expression *lib* refers to the name of the archive library and **member.o**
23757 to the member name. The application shall ensure that the member is an object file with the **.o**
23758 suffix. The modification time of the expression is the modification time for the member as kept
23759 in the archive library; see *ar*. The **.a** suffix shall refer to an archive library. The **.s2.a** rule shall be
23760 used to update a member in the library from a file with a suffix **.s2**.

23761

Internal Macros

23762

The *make* utility shall maintain five internal macros that can be used in target and inference rules. In order to clearly define the meaning of these macros, some clarification of the terms *target rule*, *inference rule*, *target*, and *prerequisite* is necessary.

23765

Target rules are specified by the user in a makefile for a particular target. Inference rules are user-specified or *make*-specified rules for a particular class of target name. Explicit prerequisites are those prerequisites specified in a makefile on target lines. Implicit prerequisites are those prerequisites that are generated when inference rules are used. Inference rules are applied to implicit prerequisites or to explicit prerequisites that do not have target rules defined for them in the makefile. Target rules are applied to targets specified in the makefile.

23771

Before any target in the makefile is updated, each of its prerequisites (both explicit and implicit) shall be updated. This shall be accomplished by recursively processing each prerequisite. Upon recursion, each prerequisite shall become a target itself. Its prerequisites in turn shall be processed recursively until a target is found that has no prerequisites, at which point the recursion stops. The recursion shall then back up, updating each target as it goes.

23776

In the definitions that follow, the word *target* refers to one of:

23777

- A target specified in the makefile
- An explicit prerequisite specified in the makefile that becomes the target when *make* processes it during recursion
- An implicit prerequisite that becomes a target when *make* processes it during recursion

23780

In the definitions that follow, the word *prerequisite* refers to one of the following:

23782

- An explicit prerequisite specified in the makefile for a particular target
- An implicit prerequisite generated as a result of locating an appropriate inference rule and corresponding file that matches the suffix of the target

23785

The five internal macros are:

23786

\$@ The \$@ shall evaluate to the full target name of the current target, or the archive filename part of a library archive target. It shall be evaluated for both target and inference rules.

23789

For example, in the .c.a inference rule, \$@ represents the out-of-date .a file to be built. Similarly, in a makefile target rule to build lib.a from file.c, \$@ represents the out-of-date lib.a.

23792

\$% The \$% macro shall be evaluated only when the current target is an archive library member of the form libname(member.o). In these cases, \$@ shall evaluate to libname and \$% shall evaluate to member.o. The \$% macro shall be evaluated for both target and inference rules.

23796

For example, in a makefile target rule to build lib.a(file.o), \$% represents file.o, as opposed to \$@, which represents lib.a.

23798

\$? The \$? macro shall evaluate to the list of prerequisites that are newer than the current target. It shall be evaluated for both target and inference rules.

23800

For example, in a makefile target rule to build prog from file1.o, file2.o, and file3.o, and where prog is not out-of-date with respect to file1.o, but is out-of-date with respect to file2.o and file3.o, \$? represents file2.o and file3.o.

23803 \$< In an inference rule, the \$< macro shall evaluate to the filename whose existence
 23804 allowed the inference rule to be chosen for the target. In the .DEFAULT rule, the \$<
 23805 macro shall evaluate to the current target name. The meaning of the \$< macro shall be
 23806 otherwise unspecified.

23807 For example, in the .c.a inference rule, \$< represents the prerequisite .c file.

23808 \$* The \$* macro shall evaluate to the current target name with its suffix deleted. It shall be
 23809 evaluated at least for inference rules.

23810 For example, in the .c.a inference rule, \$*.o represents the out-of-date .o file that
 23811 corresponds to the prerequisite .c file.

23812 Each of the internal macros has an alternative form. When an uppercase 'D' or 'F' is appended
 23813 to any of the macros, the meaning shall be changed to the *directory part* for 'D' and *filename part*
 23814 for 'F'. The directory part is the path prefix of the file without a trailing slash; for the current
 23815 directory, the directory part is '.'. When the \$? macro contains more than one prerequisite
 23816 filename, the \${?D} and \${?F} (or \${?D} and \${?F}) macros expand to a list of directory name parts
 23817 and filename parts respectively.

23818 For the target *lib(member.o)* and the s2.a rule, the internal macros shall be defined as:

23819 \$< *member.s2*
 23820 \$* *member*
 23821 \$@ *lib*
 23822 \$? *member.s2*
 23823 \$% *member.o*

23824 Default Rules

23825 The default rules for *make* shall achieve results that are the same as if the following were used.
 23826 Implementations that do not support the C-Language Development Utilities option may omit
 23827 CC, CFLAGS, YACC, YFLAGS, LEX, LFLAGS, LDFLAGS, and the .c, .y, and .l inference rules.
 23828 Implementations that do not support FORTRAN may omit FC, FFLAGS, and the .f inference
 23829 rules. Implementations may provide additional macros and rules.

23830 SPECIAL TARGETS

23831 XSI .SCCS_GET: sccs \$(SCCSFLAGS) get \$(SCCSGETFLAGS) \$@
 23832

23833 XSI .SUFFIXES: .o .c .y .l .a .sh .f .c~ .y~ .l~ .sh~ .f~

23834 MACROS

23835 MAKE=make
 23836 AR=ar
 23837 ARFLAGS=-rv
 23838 YACC=yacc
 23839 YFLAGS=
 23840 LEX=lex
 23841 LFLAGS=
 23842 LDFLAGS=
 23843 CC=c99
 23844 CFLAGS=-O
 23845 FC=fort77

```

23846      FFLAGS=-O 1
23847 XSI    GET=get
23848      GFLAGS=
23849      SCCSFLAGS=
23850      SCCSGETFLAGS=-s
23851
23852      SINGLE SUFFIX RULES
23853      .c:
23854          $(CC) $(CFLAGS) $(LDFLAGS) -o $@ $<
23855      .f:
23856          $(FC) $(FFLAGS) $(LDFLAGS) -o $@ $<
23857      .sh:
23858          cp $< $@
23859          chmod a+x $@
23860 XSI    .c~:
23861          $(GET) $(GFLAGS) -p $< > $*.c
23862          $(CC) $(CFLAGS) $(LDFLAGS) -o $@ $*.c
23863      .f~:
23864          $(GET) $(GFLAGS) -p $< > $*.f
23865          $(FC) $(FFLAGS) $(LDFLAGS) -o $@ $*.f
23866      .sh~:
23867          $(GET) $(GFLAGS) -p $< > $*.sh
23868          cp $*.sh $@
23869          chmod a+x $@
23870
23871      DOUBLE SUFFIX RULES
23872      .c.o:
23873          $(CC) $(CFLAGS) -c $<
23874      .f.o:
23875          $(FC) $(FFLAGS) -c $<
23876      .y.o:
23877          $(YACC) $(YFLAGS) $<
23878          $(CC) $(CFLAGS) -c y.tab.c
23879          rm -f y.tab.c
23880          mv y.tab.o $@
23881      .l.o:
23882          $(LEX) $(LFLAGS) $<
23883          $(CC) $(CFLAGS) -c lex.yy.c
23884          rm -f lex.yy.c
23885          mv lex.yy.o $@
23886      .y.c:
23887          $(YACC) $(YFLAGS) $<
23888          mv y.tab.c $@
23889      .l.c:
23890          $(LEX) $(LFLAGS) $<

```

```

23891           mv lex.yy.c $@
23892 XSI      .c~.o:
23893          $(GET) $(GFLAGS) -p $< > $*.c
23894          $(CC) $(CFLAGS) -c $*.c
23895      .f~.o:
23896          $(GET) $(GFLAGS) -p $< > $*.f
23897          $(FC) $(FFLAGS) -c $*.f
23898      .y~.o:
23899          $(GET) $(GFLAGS) -p $< > $*.y
23900          $(YACC) $(YFLAGS) $*.y
23901          $(CC) $(CFLAGS) -c y.tab.c
23902          rm -f y.tab.c
23903          mv y.tab.o $@
23904      .l~.o:
23905          $(GET) $(GFLAGS) -p $< > $*.l
23906          $(LEX) $(LFLAGS) $*.l
23907          $(CC) $(CFLAGS) -c lex.yy.c
23908          rm -f lex.yy.c
23909          mv lex.yy.o $@
23910      .y~.c:
23911          $(GET) $(GFLAGS) -p $< > $*.y
23912          $(YACC) $(YFLAGS) $*.y
23913          mv y.tab.c $@
23914      .l~.c:
23915          $(GET) $(GFLAGS) -p $< > $*.l
23916          $(LEX) $(LFLAGS) $*.l
23917          mv lex.yy.c $@
23918
23919      .c.a:
23920          $(CC) -c $(CFLAGS) $<
23921          $(AR) $(ARFLAGS) $@ $*.o
23922          rm -f $*.o
23923      .f.a:
23924          $(FC) -c $(FFLAGS) $<
23925          $(AR) $(ARFLAGS) $@ $*.o
23926          rm -f $*.o

```

23927 EXIT STATUS

23928 When the **-q** option is specified, the *make* utility shall exit with one of the following values:

- 23929 0 Successful completion.
- 23930 1 The target was not up-to-date.

23931 >1 An error occurred.

23932 When the **-q** option is not specified, the *make* utility shall exit with one of the following values:

- 23933 0 Successful completion.
- 23934 >0 An error occurred.

23935 CONSEQUENCES OF ERRORS

23936 Default.

23937 APPLICATION USAGE

23938 If there is a source file (such as `./source.c`) and there are two SCCS files corresponding to it
23939 (`./s.source.c` and `./SCCS/s.source.c`), on XSI-conformant systems *make* uses the SCCS file in the
23940 current directory. However, users are advised to use the underlying SCCS utilities (*admin*, *delta*,
23941 *get*, and so on) or the *sccs* utility for all source files in a given directory. If both forms are used for
23942 a given source file, future developers are very likely to be confused.

23943 It is incumbent upon portable makefiles to specify the `.POSIX` special target in order to
23944 guarantee that they are not affected by local extensions.

23945 The `-k` and `-S` options are both present so that the relationship between the command line, the
23946 `MAKEFLAGS` variable, and the makefile can be controlled precisely. If the `k` flag is passed in
23947 `MAKEFLAGS` and a command is of the form:

23948 `$(MAKE) -S foo`

23949 then the default behavior is restored for the child *make*.

23950 When the `-n` option is specified, it is always added to `MAKEFLAGS`. This allows a recursive
23951 *make -n target* to be used to see all of the action that would be taken to update *target*.

23952 Because of widespread historical practice, interpreting a '#' number sign inside a variable as
23953 the start of a comment has the unfortunate side effect of making it impossible to place a number
23954 sign in a variable, thus forbidding something like:

23955 `CFLAGS = "-D COMMENT_CHAR='#' "`

23956 Many historical *make* utilities stop chaining together inference rules when an intermediate target
23957 is nonexistent. For example, it might be possible for a *make* to determine that both `.y.c` and `.c.o`
23958 could be used to convert a `.y` to a `.o`. Instead, in this case, *make* requires the use of a `.y.o` rule.

23959 The best way to provide portable makefiles is to include all of the rules needed in the makefile
23960 itself. The rules provided use only features provided by other parts of this volume of
23961 IEEE Std 1003.1-2001. The default rules include rules for optional commands in this volume of
23962 IEEE Std 1003.1-2001. Only rules pertaining to commands that are provided are needed in an
23963 implementation's default set.

23964 Macros used within other macros are evaluated when the new macro is used rather than when
23965 the new macro is defined. Therefore:

23966 `MACRO = value1`
23967 `NEW = $(MACRO)`
23968 `MACRO = value2`

23969 `target:`
23970 `echo $(NEW)`

23971 would produce `value2` and not `value1` since `NEW` was not expanded until it was needed in the
23972 `echo` command line.

23973 Some historical applications have been known to intermix *target_name* and *macro=name* operands
23974 on the command line, expecting that all of the macros are processed before any of the targets are
23975 dealt with. Conforming applications do not do this, although some backwards-compatibility
23976 support may be included in some implementations.

23977 The following characters in filenames may give trouble: '=', ':', '^', '^', and '@'. For
23978 inference rules, the description of \$< and \$? seem similar. However, an example shows the

23979 minor difference. In a makefile containing:
23980 foo.o: foo.h
23981 if **foo.h** is newer than **foo.o**, yet **foo.c** is older than **foo.o**, the built-in rule to make **foo.o** from
23982 **foo.c** is used, with \$< equal to **foo.c** and \$? equal to **foo.h**. If **foo.c** is also newer than **foo.o**, \$< is
23983 equal to **foo.c** and \$? is equal to **foo.h** **foo.c**.

23984 EXAMPLES

- 23985 1. The following command:
23986 make
23987 makes the first target found in the makefile.
- 23988 2. The following command:
23989 make junk
23990 makes the target **junk**.
- 23991 3. The following makefile says that **pgm** depends on two files, **a.o** and **b.o**, and that they in
23992 turn depend on their corresponding source files (**a.c** and **b.c**), and a common file **incl.h**:

```
23993     pgm: a.o b.o  
23994         c99 a.o b.o -o pgm  
23995     a.o: incl.h a.c  
23996         c99 -c a.c  
23997     b.o: incl.h b.c  
23998         c99 -c b.c
```

- 23999 4. An example for making optimized .o files from .c files is:

```
24000     .c.o:  
24001         c99 -c -O $*.c
```

24002 or:

```
24003     .c.o:  
24004         c99 -c -O $<
```

- 24005 5. The most common use of the archive interface follows. Here, it is assumed that the source
24006 files are all C-language source:

```
24007     lib: lib(file1.o) lib(file2.o) lib(file3.o)  
24008         @echo lib is now up-to-date
```

24009 The **.c.a** rule is used to make **file1.o**, **file2.o**, and **file3.o** and insert them into **lib**.

24010 The treatment of escaped <newline>s throughout the makefile is historical practice. For
24011 example, the inference rule:

```
24012     .c.o\  
24013     :  
24014         works, and the macro:  
24015         f= bar baz\  
24016             biz  
24017         a:  
24018             echo ==$f==
```

24019 echoes " ==bar baz biz==".
24020 If \$? were:
24021 /usr/include/stdio.h /usr/include/unistd.h foo.h
24022 then \$(?D) would be:
24023 /usr/include /usr/include .
24024 and \$(?F) would be:
24025 stdio.h unistd.h foo.h
24026 6. The contents of the built-in rules can be viewed by running:
24027 make -p -f /dev/null 2>/dev/null

24028 RATIONALE

24029 The *make* utility described in this volume of IEEE Std 1003.1-2001 is intended to provide the
24030 means for changing portable source code into executables that can be run on an
24031 IEEE Std 1003.1-2001-conforming system. It reflects the most common features present in
24032 System V and BSD *makes*.

24033 Historically, the *make* utility has been an especially fertile ground for vendor and research
24034 organization-specific syntax modifications and extensions. Examples include:

- 24035 • Syntax supporting parallel execution (such as from various multi-processor vendors, GNU,
24036 and others)
- 24037 • Additional “operators” separating targets and their prerequisites (System V, BSD, and
24038 others)
- 24039 • Specifying that command lines containing the strings “\${MAKE}” and “\$(MAKE)” are
24040 executed when the **-n** option is specified (GNU and System V)
- 24041 • Modifications of the meaning of internal macros when referencing libraries (BSD and others)
- 24042 • Using a single instance of the shell for all of the command lines of the target (BSD and others)
- 24043 • Allowing spaces as well as tabs to delimit command lines (BSD)
- 24044 • Adding C preprocessor-style “include” and “ifdef” constructs (System V, GNU, BSD, and
24045 others)
- 24046 • Remote execution of command lines (Sprite and others)
- 24047 • Specifying additional special targets (BSD, System V, and most others)

24048 Additionally, many vendors and research organizations have rethought the basic concepts of
24049 *make*, creating vastly extended, as well as completely new, syntaxes. Each of these versions of
24050 *make* fulfills the needs of a different community of users; it is unreasonable for this volume of
24051 IEEE Std 1003.1-2001 to require behavior that would be incompatible (and probably inferior) to
24052 historical practice for such a community.

24053 In similar circumstances, when the industry has enough sufficiently incompatible formats as to
24054 make them irreconcilable, this volume of IEEE Std 1003.1-2001 has followed one or both of two
24055 courses of action. Commands have been renamed (*cksum*, *echo*, and *pax*) and/or command line
24056 options have been provided to select the desired behavior (*grep*, *od*, and *pax*).

24057 Because the syntax specified for the *make* utility is, by and large, a subset of the syntaxes
24058 accepted by almost all versions of *make*, it was decided that it would be counter-productive to
24059 change the name. And since the makefile itself is a basic unit of portability, it would not be

24060 completely effective to reserve a new option letter, such as *make -P*, to achieve the portable
24061 behavior. Therefore, the special target **.POSIX** was added to the makefile, allowing users to
24062 specify “standard” behavior. This special target does not preclude extensions in the *make* utility,
24063 nor does it preclude such extensions being used by the makefile specifying the target; it does,
24064 however, preclude any extensions from being applied that could alter the behavior of previously
24065 valid syntax; such extensions must be controlled via command line options or new special
24066 targets. It is incumbent upon portable makefiles to specify the **.POSIX** special target in order to
24067 guarantee that they are not affected by local extensions.

24068 The portable version of *make* described in this reference page is not intended to be the state-of-
24069 the-art software generation tool and, as such, some newer and more leading-edge features have
24070 not been included. An attempt has been made to describe the portable makefile in a manner that
24071 does not preclude such extensions as long as they do not disturb the portable behavior described
24072 here.

24073 When the **-n** option is specified, it is always added to **MAKEFLAGS**. This allows a recursive
24074 *make -n target* to be used to see all of the action that would be taken to update *target*.

24075 The definition of **MAKEFLAGS** allows both the System V letter string and the BSD command line
24076 formats. The two formats are sufficiently different to allow implementations to support both
24077 without ambiguity.

24078 Early proposals stated that an “unquoted” number sign was treated as the start of a comment.
24079 The *make* utility does not pay any attention to quotes. A number sign starts a comment
24080 regardless of its surroundings.

24081 The text about “other implementation-defined pathnames may also be tried” in addition to
24082 **./makefile** and **./Makefile** is to allow such extensions as **SCCS/s.Makefile** and other variations.
24083 It was made an implementation-defined requirement (as opposed to unspecified behavior) to
24084 highlight surprising implementations that might select something unexpected like
24085 **/etc/Makefile**. XSI-conformant systems also try **./s.makefile**, **SCCS/s.makefile**, **./s.Makefile**,
24086 and **SCCS/s.Makefile**.

24087 Early proposals contained the macro **NPROC** as a means of specifying that *make* should use *n*
24088 processes to do the work required. While this feature is a valuable extension for many systems, it
24089 is not common usage and could require other non-trivial extensions to makefile syntax. This
24090 extension is not required by this volume of IEEE Std 1003.1-2001, but could be provided as a
24091 compatible extension. The macro **PARALLEL** is used by some historical systems with essentially
24092 the same meaning (but without using a name that is a common system limit value). It is
24093 suggested that implementors recognize the existing use of **NPROC** and/or **PARALLEL** as
24094 extensions to *make*.

24095 The default rules are based on System V. The default **CC=** value is **c99** instead of **cc** because this
24096 volume of IEEE Std 1003.1-2001 does not standardize the utility named **cc**. Thus, every
24097 conforming application would be required to define **CC=c99** to expect to run. There is no
24098 advantage conferred by the hope that the makefile might hit the “preferred” compiler because
24099 this cannot be guaranteed to work. Also, since the portable makescript can only use the **c99**
24100 options, no advantage is conferred in terms of what the script can do. It is a quality-of-
24101 implementation issue as to whether **c99** is as valuable as **cc**.

24102 The **-d** option to *make* is frequently used to produce debugging information, but is too
24103 implementation-defined to add to this volume of IEEE Std 1003.1-2001.

24104 The **-p** option is not passed in **MAKEFLAGS** on most historical implementations and to change
24105 this would cause many implementations to break without sufficiently increased portability.

24106 Commands that begin with a plus sign ('+') are executed even if the **-n** option is present. Based
24107 on the GNU version of *make*, the behavior of **-n** when the plus-sign prefix is encountered has
24108 been extended to apply to **-q** and **-t** as well. However, the System V convention of forcing
24109 command execution with **-n** when the command line of a target contains either of the strings
24110 "**\$(MAKE)**" or " **\${MAKE}**" has not been adopted. This functionality appeared in early
24111 proposals, but the danger of this approach was pointed out with the following example of a
24112 portion of a makefile:

24113 **subdir:**
24114 cd subdir; rm all_the_files; \$(MAKE)

24115 The loss of the System V behavior in this case is well-balanced by the safety afforded to other
24116 makefiles that were not aware of this situation. In any event, the command line plus-sign prefix
24117 can provide the desired functionality.

24118 The double colon in the target rule format is supported in BSD systems to allow more than one
24119 target line containing the same target name to have commands associated with it. Since this is
24120 not functionality described in the SVID or XPG3 it has been allowed as an extension, but not
24121 mandated.

24122 The default rules are provided with text specifying that the built-in rules shall be the same as if
24123 the listed set were used. The intent is that implementations should be able to use the rules
24124 without change, but will be allowed to alter them in ways that do not affect the primary
24125 behavior.

24126 The best way to provide portable makefiles is to include all of the rules needed in the makefile
24127 itself. The rules provided use only features provided by other portions of this volume of
24128 IEEE Std 1003.1-2001. The default rules include rules for optional commands in this volume of
24129 IEEE Std 1003.1-2001. Only rules pertaining to commands that are provided are needed in the
24130 default set of an implementation.

24131 One point of discussion was whether to drop the default rules list from this volume of
24132 IEEE Std 1003.1-2001. They provide convenience, but do not enhance portability of applications.
24133 The prime benefit is in portability of users who wish to type *make command* and have the
24134 command build from a **command.c** file.

24135 The historical *MAKESHELL* feature was omitted. In some implementations it is used to let a user
24136 override the shell to be used to run *make* commands. This was confusing; for a portable *make*, the
24137 shell should be chosen by the makefile writer or specified on the *make* command line and not by
24138 a user running *make*.

24139 The *make* utilities in most historical implementations process the prerequisites of a target in left-
24140 to-right order, and the makefile format requires this. It supports the standard idiom used in
24141 many makefiles that produce *yacc* programs; for example:

24142 **foo: y.tab.o lex.o main.o**
24143 \$(CC) \$(CFLAGS) -o \$@ t.tab.o lex.o main.o

24144 In this example, if *make* chose any arbitrary order, the **lex.o** might not be made with the correct
24145 **y.tab.h**. Although there may be better ways to express this relationship, it is widely used
24146 historically. Implementations that desire to update prerequisites in parallel should require an
24147 explicit extension to *make* or the makefile format to accomplish it, as described previously.

24148 The algorithm for determining a new entry for target rules is partially unspecified. Some
24149 historical *makes* allow blank, empty, or comment lines within the collection of commands
24150 marked by leading <tab>s. A conforming makefile must ensure that each command starts with
24151 a <tab>, but implementations are free to ignore blank, empty, and comment lines without
24152 triggering the start of a new entry.

24153 The ASYNCHRONOUS EVENTS section includes having SIGTERM and SIGHUP, along with
24154 the more traditional SIGINT and SIGQUIT, remove the current target unless directed not to do
24155 so. SIGTERM and SIGHUP were added to parallel other utilities that have historically cleaned
24156 up their work as a result of these signals. When *make* receives any signal other than SIGQUIT, it
24157 is required to resend itself the signal it received so that it exits with a status that reflects the
24158 signal. The results from SIGQUIT are partially unspecified because, on systems that create **core**
24159 files upon receipt of SIGQUIT, the **core** from *make* would conflict with a **core** file from the
24160 command that was running when the SIGQUIT arrived. The main concern was to prevent
24161 damaged files from appearing up-to-date when *make* is rerun.

24162 The **.PRECIOUS** special target was extended to affect all targets globally (by specifying no
24163 prerequisites). The **.IGNORE** and **.SILENT** special targets were extended to allow prerequisites;
24164 it was judged to be more useful in some cases to be able to turn off errors or echoing for a list of
24165 targets than for the entire makefile. These extensions to *make* in System V were made to match
24166 historical practice from the BSD *make*.

24167 Macros are not exported to the environment of commands to be run. This was never the case in
24168 any historical *make* and would have serious consequences. The environment is the same as the
24169 environment to *make* except that **MAKEFLAGS** and macros defined on the *make* command line
24170 are added.

24171 Some implementations do not use *system()* for all command lines, as required by the portable
24172 makefile format; as a performance enhancement, they select lines without shell metacharacters
24173 for direct execution by *execve()*. There is no requirement that *system()* be used specifically, but
24174 merely that the same results be achieved. The metacharacters typically used to bypass the direct
24175 *execve()* execution have been any of:

24176 = | ^ () ; & < > * ? [] : \$ ' ' " \ \n

24177 The default in some advanced versions of *make* is to group all the command lines for a target and
24178 execute them using a single shell invocation; the System V method is to pass each line
24179 individually to a separate shell. The single-shell method has the advantages in performance and
24180 the lack of a requirement for many continued lines. However, converting to this newer method
24181 has caused portability problems with many historical makefiles, so the behavior with the POSIX
24182 makefile is specified to be the same as that of System V. It is suggested that the special target
24183 **.ONESHELL** be used as an implementation extension to achieve the single-shell grouping for a
24184 target or group of targets.

24185 Novice users of *make* have had difficulty with the historical need to start commands with a
24186 <tab>. Since it is often difficult to discern differences between <tab>s and <space>s on terminals
24187 or printed listings, confusing bugs can arise. In early proposals, an attempt was made to correct
24188 this problem by allowing leading <blank>s instead of <tab>s. However, implementors reported
24189 many makefiles that failed in subtle ways following this change, and it is difficult to implement
24190 a *make* that unambiguously can differentiate between macro and command lines. There is
24191 extensive historical practice of allowing leading spaces before macro definitions. Forcing macro
24192 lines into column 1 would be a significant backwards-compatibility problem for some makefiles.
24193 Therefore, historical practice was restored.

24194 The System V INCLUDE feature was considered, but not included. This would treat a line that
24195 began in the first column and contained INCLUDE <filename> as an indication to read <filename>
24196 at that point in the makefile. This is difficult to use in a portable way, and it raises concerns
24197 about nesting levels and diagnostics. System V, BSD, GNU, and others have used different
24198 methods for including files.

24199 The System V dynamic dependency feature was not included. It would support:

24200 cat: \$\$@.c
24201 that would expand to;
24202 cat: cat.c
24203 This feature exists only in the new version of System V *make* and, while useful, is not in wide
24204 usage. This means that macros are expanded twice for prerequisites: once at makefile parse time
24205 and once at target update time.
24206 Consideration was given to adding metarules to the POSIX *make*. This would make %.o: %.c the
24207 same as .c.o:. This is quite useful and available from some vendors, but it would cause too many
24208 changes to this *make* to support. It would have introduced rule chaining and new substitution
24209 rules. However, the rules for target names have been set to reserve the '%' and '"' characters.
24210 These are traditionally used to implement metarules and quoting of target names, respectively.
24211 Implementors are strongly encouraged to use these characters only for these purposes.
24212 A request was made to extend the suffix delimiter character from a period to any character. The
24213 metarules feature in newer *makes* solves this problem in a more general way. This volume of
24214 IEEE Std 1003.1-2001 is staying with the more conservative historical definition.
24215 The standard output format for the -p option is not described because it is primarily a
24216 debugging option and because the format is not generally useful to programs. In historical
24217 implementations the output is not suitable for use in generating makefiles. The -p format has
24218 been variable across historical implementations. Therefore, the definition of -p was only to
24219 provide a consistently named option for obtaining *make* script debugging information.
24220 Some historical implementations have not cleared the suffix list with -r.
24221 Implementations should be aware that some historical applications have intermixed *target_name*
24222 and *macro=value* operands on the command line, expecting that all of the macros are processed
24223 before any of the targets are dealt with. Conforming applications do not do this, but some
24224 backwards-compatibility support may be warranted.
24225 Empty inference rules are specified with a semicolon command rather than omitting all
24226 commands, as described in an early proposal. The latter case has no traditional meaning and is
24227 reserved for implementation extensions, such as in GNU *make*.

24228 **FUTURE DIRECTIONS**
24229 None.

24230 **SEE ALSO**
24231 Chapter 2 (on page 29), *ar*, *c99*, *get*, *lex*, *sccs*, *sh*, *yacc*, the System Interfaces volume of
24232 IEEE Std 1003.1-2001, *exec*, *system()*

24233 **CHANGE HISTORY**
24234 First released in Issue 2.

24235 **Issue 5**
24236 The FUTURE DIRECTIONS section is added.

24237 **Issue 6**
24238 This utility is marked as part of the Software Development Utilities option.
24239 The Open Group Corrigendum U029/1 is applied, correcting a typographical error in the
24240 SPECIAL TARGETS section.
24241 In the ENVIRONMENT VARIABLES section, the *PROJECTDIR* description is updated from
24242 "otherwise, the home directory of a user of that name is examined" to "otherwise, the value of
24243 *PROJECTDIR* is treated as a user name and that user's initial working directory is examined".

- 24244 It is specified whether the command line is related to the makefile or to the *make* command, and
24245 the macro processing rules are updated to align with the IEEE P1003.2b draft standard.
- 24246 The normative text is reworded to avoid use of the term “must” for application requirements.
- 24247 PASC Interpretation 1003.2 #193 is applied.

24248 NAME

24249 man — display system documentation

24250 SYNOPSIS

24251 man [-k] name...

24252 DESCRIPTION

24253 The *man* utility shall write information about each of the *name* operands. If *name* is the name of a standard utility, *man* at a minimum shall write a message describing the syntax used by the standard utility, its options, and operands. If more information is available, the *man* utility shall provide it in an implementation-defined manner.

24257 An implementation may provide information for values of *name* other than the standard utilities. Standard utilities that are listed as optional and that are not supported by the implementation either shall cause a brief message indicating that fact to be displayed or shall cause a full display of information as described previously.

24261 OPTIONS

24262 The *man* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

24264 The following option shall be supported:

24265 **-k** Interpret *name* operands as keywords to be used in searching a utilities summary database that contains a brief purpose entry for each standard utility and write lines from the summary database that match any of the keywords. The keyword search shall produce results that are the equivalent of the output of the following command:

```
24269 grep -Ei '
24270   name
24271   name
24272   ...
24273   ' summary-database
```

24274 This assumes that the *summary-database* is a text file with a single entry per line; this organization is not required and the example using *grep -Ei* is merely illustrative of the type of search intended. The purpose entry to be included in the database shall consist of a terse description of the purpose of the utility.

24278 OPERANDS

24279 The following operand shall be supported:

24280 *name* A keyword or the name of a standard utility. When **-k** is not specified and *name* does not represent one of the standard utilities, the results are unspecified.

24282 STDIN

24283 Not used.

24284 INPUT FILES

24285 None.

24286 ENVIRONMENT VARIABLES

24287 The following environment variables shall affect the execution of *man*:

24288 **LANG** Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

24292	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
24294	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and in the summary database). The value of <i>LC_CTYPE</i> need not affect the format of the information written about the <i>name</i> operands.
24298	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.
24302 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
24303	<i>PAGER</i>	Determine an output filtering command for writing the output to a terminal. Any string acceptable as a <i>command_string</i> operand to the <i>sh -c</i> command shall be valid. When standard output is a terminal device, the reference page output shall be piped through the command. If the <i>PAGER</i> variable is null or not set, the command shall be either <i>more</i> or another paginator utility documented in the system documentation.
24309	ASYNCHRONOUS EVENTS	
24310		Default.
24311	STDOUT	
24312		The <i>man</i> utility shall write text describing the syntax of the utility <i>name</i> , its options and its operands, or, when -k is specified, lines from the summary database. The format of this text is implementation-defined.
24315	STDERR	
24316		The standard error shall be used only for diagnostic messages.
24317	OUTPUT FILES	
24318		None.
24319	EXTENDED DESCRIPTION	
24320		None.
24321	EXIT STATUS	
24322		The following exit values shall be returned:
24323		0 Successful completion.
24324		>0 An error occurred.
24325	CONSEQUENCES OF ERRORS	
24326		Default.
24327	APPLICATION USAGE	
24328		None.
24329	EXAMPLES	
24330		None.
24331	RATIONALE	
24332		It is recognized that the <i>man</i> utility is only of minimal usefulness as specified. The opinion of the standard developers was strongly divided as to how much or how little information <i>man</i> should be required to provide. They considered, however, that the provision of some portable way of accessing documentation would aid user portability. The arguments against a fuller

24336 specification were:

- 24337 • Large quantities of documentation should not be required on a system that does not have
24338 excess disk space.
- 24339 • The current manual system does not present information in a manner that greatly aids user
24340 portability.
- 24341 • A “better help system” is currently an area in which vendors feel that they can add value to
24342 their POSIX implementations.

24343 The **-f** option was considered, but due to implementation differences, it was not included in this
24344 volume of IEEE Std 1003.1-2001.

24345 The description was changed to be more specific about what has to be displayed for a utility.
24346 The standard developers considered it insufficient to allow a display of only the synopsis
24347 without giving a short description of what each option and operand does.

24348 The “purpose” entry to be included in the database can be similar to the section title (less the
24349 numeric prefix) from this volume of IEEE Std 1003.1-2001 for each utility. These titles are similar
24350 to those used in historical systems for this purpose.

24351 See *mailx* for rationale concerning the default paginator.

24352 The caveat in the *LC_CTYPE* description was added because it is not a requirement that an
24353 implementation provide reference pages for all of its supported locales on each system;
24354 changing *LC_CTYPE* does not necessarily translate the reference page into another language.
24355 This is equivalent to the current state of *LC_MESSAGES* in IEEE Std 1003.1-2001—locale-specific
24356 messages are not yet a requirement.

24357 The historical *MANPATH* variable is not included in POSIX because no attempt is made to
24358 specify naming conventions for reference page files, nor even to mandate that they are files at
24359 all. On some implementations they could be a true database, a hypertext file, or even fixed
24360 strings within the *man* executable. The standard developers considered the portability of
24361 reference pages to be outside their scope of work. However, users should be aware that
24362 *MANPATH* is implemented on a number of historical systems and that it can be used to tailor
24363 the search pattern for reference pages from the various categories (utilities, functions, file
24364 formats, and so on) when the system administrator reveals the location and conventions for
24365 reference pages on the system.

24366 The keyword search can rely on at least the text of the section titles from these utility
24367 descriptions, and the implementation may add more keywords. The term “section titles” refers
24368 to the strings such as:

24369 man – Display system documentation
24370 ps – Report process status

24371 FUTURE DIRECTIONS

24372 None.

24373 SEE ALSO

24374 *more*

24375 CHANGE HISTORY

24376 First released in Issue 4.

24377 **Issue 5**

24378 The FUTURE DIRECTIONS section is added.

24379 NAME

24380 mesg — permit or deny messages

24381 SYNOPSIS

24382 UP mesg [y|n]

24383

24384 DESCRIPTION

24385 The *mesg* utility shall control whether other users are allowed to send messages via *write*, *talk*, or
24386 other utilities to a terminal device. The terminal device affected shall be determined by searching
24387 for the first terminal in the sequence of devices associated with standard input, standard output,
24388 and standard error, respectively. With no arguments, *mesg* shall report the current state without
24389 changing it. Processes with appropriate privileges may be able to send messages to the terminal
24390 independent of the current state.

24391 OPTIONS

24392 None.

24393 OPERANDS

24394 The following operands shall be supported in the POSIX locale:

24395 *y* Grant permission to other users to send messages to the terminal device.

24396 *n* Deny permission to other users to send messages to the terminal device.

24397 STDIN

24398 Not used.

24399 INPUT FILES

24400 None.

24401 ENVIRONMENT VARIABLES

24402 The following environment variables shall affect the execution of *mesg*:

24403 *LANG* Provide a default value for the internationalization variables that are unset or null.
24404 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
24405 Internationalization Variables for the precedence of internationalization variables
24406 used to determine the values of locale categories.)

24407 *LC_ALL* If set to a non-empty string value, override the values of all the other
24408 internationalization variables.

24409 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
24410 characters (for example, single-byte as opposed to multi-byte characters in
24411 arguments).

24412 *LC_MESSAGES*

24413 Determine the locale that should be used to affect the format and contents of
24414 diagnostic messages written (by *mesg*) to standard error.

24415 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

24416 ASYNCHRONOUS EVENTS

24417 Default.

24418 STDOUT

24419 If no operand is specified, *mesg* shall display the current terminal state in an unspecified format.

24420 STDRERR

24421 The standard error shall be used only for diagnostic messages.

24422 OUTPUT FILES

24423 None.

24424 EXTENDED DESCRIPTION

24425 None.

24426 EXIT STATUS

24427 The following exit values shall be returned:

24428 0 Receiving messages is allowed.

24429 1 Receiving messages is not allowed.

24430 >1 An error occurred.

24431 CONSEQUENCES OF ERRORS

24432 Default.

24433 APPLICATION USAGE

24434 The mechanism by which the message status of the terminal is changed is unspecified.
24435 Therefore, unspecified actions may cause the status of the terminal to change after *mesg* has
24436 successfully completed. These actions may include, but are not limited to: another invocation of
24437 the *mesg* utility, login procedures; invocation of the *stty* utility, invocation of the *chmod* utility or
24438 *chmod()* function, and so on.

24439 EXAMPLES

24440 None.

24441 RATIONALE

24442 The terminal changed by *mesg* is that associated with the standard input, output, or error, rather
24443 than the controlling terminal for the session. This is because users logged in more than once
24444 should be able to change any of their login terminals without having to stop the job running in
24445 those sessions. This is not a security problem involving the terminals of other users because
24446 appropriate privileges would be required to affect the terminal of another user.

24447 The method of checking each of the first three file descriptors in sequence until a terminal is
24448 found was adopted from System V.

24449 The file */dev/tty* is not specified for the terminal device because it was thought to be too
24450 restrictive. Typical environment changes for the *n* operand are that write permissions are
24451 removed for *others* and *group* from the appropriate device. It was decided to leave the actual
24452 description of what is done as unspecified because of potential differences between
24453 implementations.

24454 The format for standard output is unspecified because of differences between historical
24455 implementations. This output is generally not useful to shell scripts (they can use the exit
24456 status), so exact parsing of the output is unnecessary.

24457 FUTURE DIRECTIONS

24458 None.

24459 SEE ALSO

24460 *talk*, *write*

24461 CHANGE HISTORY

24462 First released in Issue 2.

24463 Issue 6

24464 This utility is marked as part of the User Portability Utilities option.

24465 **NAME**

24466 mkdir — make directories

24467 **SYNOPSIS**

24468 mkdir [-p][-m mode] dir...

24469 **DESCRIPTION**24470 The *mkdir* utility shall create the directories specified by the operands, in the order specified.24471 For each *dir* operand, the *mkdir* utility shall perform actions equivalent to the *mkdir()* function
24472 defined in the System Interfaces volume of IEEE Std 1003.1-2001, called with the following
24473 arguments:

- 24474 1. The *dir* operand is used as the *path* argument.
- 24475 2. The value of the bitwise-inclusive OR of S_IRWXU, S_IRWXG, and S_IRWXO is used as
24476 the *mode* argument. (If the **-m** option is specified, the *mode* option-argument overrides this
24477 default.)

24478 **OPTIONS**24479 The *mkdir* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
24480 12.2, Utility Syntax Guidelines.

24481 The following options shall be supported:

24482 **-m mode** Set the file permission bits of the newly-created directory to the specified *mode*
24483 value. The *mode* option-argument shall be the same as the *mode* operand defined
24484 for the *chmod* utility. In the *symbolic_mode* strings, the *op* characters '+' and '-'
24485 shall be interpreted relative to an assumed initial mode of a=rwx; '+' shall add
24486 permissions to the default mode, '-' shall delete permissions from the default
24487 mode.24488 **-p** Create any missing intermediate pathname components.24489 For each *dir* operand that does not name an existing directory, effects equivalent to
24490 those caused by the following command shall occur:24491

```
mkdir -p -m $(umask -S),u+wx $(dirname dir) &&
24492       mkdir [-m mode] dir
```

24493 where the **-m mode** option represents that option supplied to the original
24494 invocation of *mkdir*, if any.24495 Each *dir* operand that names an existing directory shall be ignored without error.24496 **OPERANDS**

24497 The following operand shall be supported:

24498 **dir** A pathname of a directory to be created.24499 **STDIN**

24500 Not used.

24501 **INPUT FILES**

24502 None.

24503 **ENVIRONMENT VARIABLES**24504 The following environment variables shall affect the execution of *mkdir*:24505 **LANG** Provide a default value for the internationalization variables that are unset or null.
24506 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
24507 Internationalization Variables for the precedence of internationalization variables

24508		used to determine the values of locale categories.)
24509	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
24511	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
24514	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
24517 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
24518	ASYNCHRONOUS EVENTS	
24519		Default.
24520	STDOUT	
24521		Not used.
24522	STDERR	
24523		The standard error shall be used only for diagnostic messages.
24524	OUTPUT FILES	
24525		None.
24526	EXTENDED DESCRIPTION	
24527		None.
24528	EXIT STATUS	
24529		The following exit values shall be returned:
24530	0	All the specified directories were created successfully or the -p option was specified and all the specified directories now exist.
24531		
24532	>0	An error occurred.
24533	CONSEQUENCES OF ERRORS	
24534		Default.
24535	APPLICATION USAGE	
24536		The default file mode for directories is <i>a=rwx</i> (777 on most systems) with selected permissions removed in accordance with the file mode creation mask. For intermediate pathname components created by <i>mkdir</i> , the mode is the default modified by <i>u+wx</i> so that the subdirectories can always be created regardless of the file mode creation mask; if different ultimate permissions are desired for the intermediate directories, they can be changed afterwards with <i>chmod</i> .
24537		
24538		
24539		
24540		
24541		
24542		Note that some of the requested directories may have been created even if an error occurs.
24543	EXAMPLES	
24544		None.
24545	RATIONALE	
24546		The System V -m option was included to control the file mode.
24547		The System V -p option was included to create any needed intermediate directories and to complement the functionality provided by <i>rmdir</i> for removing directories in the path prefix as they become empty. Because no error is produced if any path component already exists, the -p option is also useful to ensure that a particular directory exists.
24548		
24549		
24550		

24551 The functionality of *mkdir* is described substantially through a reference to the *mkdir()* function
24552 in the System Interfaces volume of IEEE Std 1003.1-2001. For example, by default, the mode of
24553 the directory is affected by the file mode creation mask in accordance with the specified
24554 behavior of the *mkdir()* function. In this way, there is less duplication of effort required for
24555 describing details of the directory creation.

24556 **FUTURE DIRECTIONS**

24557 None.

24558 **SEE ALSO**

24559 *chmod*, *rm*, *rmdir*, *umask*, the System Interfaces volume of IEEE Std 1003.1-2001, *mkdir()*

24560 **CHANGE HISTORY**

24561 First released in Issue 2.

24562 **Issue 5**

24563 The FUTURE DIRECTIONS section is added.

24564 NAME

24565 mkfifo — make FIFO special files

24566 SYNOPSIS

24567 mkfifo [-m mode] file...

24568 DESCRIPTION

24569 The *mkfifo* utility shall create the FIFO special files specified by the operands, in the order
24570 specified.

24571 For each *file* operand, the *mkfifo* utility shall perform actions equivalent to the *mkfifo()* function
24572 defined in the System Interfaces volume of IEEE Std 1003.1-2001, called with the following
24573 arguments:

- 24574 1. The *file* operand is used as the *path* argument.
- 24575 2. The value of the bitwise-inclusive OR of S_IRUSR, S_IWUSR, S_IRGRP, S_IWGRP,
24576 S_IROTH, and S_IWOTH is used as the *mode* argument. (If the **-m** option is specified, the
24577 value of the *mkfifo()* *mode* argument is unspecified, but the FIFO shall at no time have
24578 permissions less restrictive than the **-m** *mode* option-argument.)

24579 OPTIONS

24580 The *mkfifo* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
24581 12.2, Utility Syntax Guidelines.

24582 The following option shall be supported:

24583 **-m mode** Set the file permission bits of the newly-created FIFO to the specified *mode* value.
24584 The *mode* option-argument shall be the same as the *mode* operand defined for the
24585 *chmod* utility. In the *symbolic_mode* strings, the *op* characters '+' and '-' shall be
24586 interpreted relative to an assumed initial mode of *a=rw*.

24587 OPERANDS

24588 The following operand shall be supported:

24589 *file* A pathname of the FIFO special file to be created.

24590 STDIN

24591 Not used.

24592 INPUT FILES

24593 None.

24594 ENVIRONMENT VARIABLES

24595 The following environment variables shall affect the execution of *mkfifo*:

24596 **LANG** Provide a default value for the internationalization variables that are unset or null.
24597 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
24598 Internationalization Variables for the precedence of internationalization variables
24599 used to determine the values of locale categories.)

24600 **LC_ALL** If set to a non-empty string value, override the values of all the other
24601 internationalization variables.

24602 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
24603 characters (for example, single-byte as opposed to multi-byte characters in
24604 arguments).

24605 *LC_MESSAGES*

24606 Determine the locale that should be used to affect the format and contents of
24607 diagnostic messages written to standard error.

24608 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

24609 **ASYNCHRONOUS EVENTS**

24610 Default.

24611 **STDOUT**

24612 Not used.

24613 **STDERR**

24614 The standard error shall be used only for diagnostic messages.

24615 **OUTPUT FILES**

24616 None.

24617 **EXTENDED DESCRIPTION**

24618 None.

24619 **EXIT STATUS**

24620 The following exit values shall be returned:

24621 0 All the specified FIFO special files were created successfully.

24622 >0 An error occurred.

24623 **CONSEQUENCES OF ERRORS**

24624 Default.

24625 **APPLICATION USAGE**

24626 None.

24627 **EXAMPLES**

24628 None.

24629 **RATIONALE**

24630 This utility was added to permit shell applications to create FIFO special files.

24631 The **-m** option was added to control the file mode, for consistency with the similar functionality provided by the *mkdir* utility.

24633 Early proposals included a **-p** option similar to the *mkdir -p* option that created intermediate directories leading up to the FIFO specified by the final component. This was removed because it is not commonly needed and is not common practice with similar utilities.

24636 The functionality of *mkfifo* is described substantially through a reference to the *mkfifo()* function in the System Interfaces volume of IEEE Std 1003.1-2001. For example, by default, the mode of the FIFO file is affected by the file mode creation mask in accordance with the specified behavior of the *mkfifo()* function. In this way, there is less duplication of effort required for describing details of the file creation.

24641 **FUTURE DIRECTIONS**

24642 None.

24643 **SEE ALSO**

24644 *chmod*, *umask*, the System Interfaces volume of IEEE Std 1003.1-2001, *mkfifo()*

24645 **CHANGE HISTORY**

24646 First released in Issue 3.

24647 NAME

24648 more — display files on a page-by-page basis

24649 SYNOPSIS

24650 UP `more [-ceisu][-n number][-p command][-t tagstring][file ...]`

24651

24652 DESCRIPTION

24653 The *more* utility shall read files and either write them to the terminal on a page-by-page basis or
24654 filter them to standard output. If standard output is not a terminal device, all input files shall be
24655 copied to standard output in their entirety, without modification, except as specified for the **-s**
24656 option. If standard output is a terminal device, the files shall be written a number of lines (one
24657 screenful) at a time under the control of user commands. See the EXTENDED DESCRIPTION
24658 section.

24659 Certain block-mode terminals do not have all the capabilities necessary to support the complete
24660 *more* definition; they are incapable of accepting commands that are not terminated with a
24661 <newline>. Implementations that support such terminals shall provide an operating mode to
24662 *more* in which all commands can be terminated with a <newline> on those terminals. This mode:

- 24663 • Shall be documented in the system documentation
- 24664 • Shall, at invocation, inform the user of the terminal deficiency that requires the <newline>
24665 usage and provide instructions on how this warning can be suppressed in future invocations
- 24666 • Shall not be required for implementations supporting only fully capable terminals
- 24667 • Shall not affect commands already requiring <newline>s
- 24668 • Shall not affect users on the capable terminals from using *more* as described in this volume of
24669 IEEE Std 1003.1-2001

24670 OPTIONS

24671 The *more* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
24672 12.2, Utility Syntax Guidelines.

24673 The following options shall be supported:

- 24674 **-c** If a screen is to be written that has no lines in common with the current screen, or
24675 *more* is writing its first screen, *more* shall not scroll the screen, but instead shall
24676 redraw each line of the screen in turn, from the top of the screen to the bottom. In
24677 addition, if *more* is writing its first screen, the screen shall be cleared. This option
24678 may be silently ignored on devices with insufficient terminal capabilities.
- 24679 **-e** By default, *more* shall exit immediately after writing the last line of the last file in
24680 the argument list. If the **-e** option is specified:
 - 24681 1. If there is only a single file in the argument list and that file was completely
24682 displayed on a single screen, *more* shall exit immediately after writing the last
24683 line of that file.
 - 24684 2. Otherwise, *more* shall exit only after reaching end-of-file on the last file in the
24685 argument list twice without an intervening operation. See the EXTENDED
24686 DESCRIPTION section.
- 24687 **-i** Perform pattern matching in searches without regard to case; see the Base
24688 Definitions volume of IEEE Std 1003.1-2001, Section 9.2, Regular Expression
24689 General Requirements.

- 24690 **-n number** Specify the number of lines per screenful. The *number* argument is a positive
24691 decimal integer. The **-n** option shall override any values obtained from any other
24692 source.
- 24693 **-p command** Each time a screen from a new file is displayed or redisplayed (including as a
24694 result of *more* commands; for example, **:p**), execute the *more* command(s) in the
24695 command arguments in the order specified, as if entered by the user after the first
24696 screen has been displayed. No intermediate results shall be displayed (that is, if the
24697 command is a movement to a screen different from the normal first screen, only
24698 the screen resulting from the command shall be displayed.) If any of the
24699 commands fail for any reason, an informational message to this effect shall be
24700 written, and no further commands specified using the **-p** option shall be executed
24701 for this file.
- 24702 **-s** Behave as if consecutive empty lines were a single empty line.
- 24703 **-t tagstring** Write the screenful of the file containing the tag named by the *tagstring* argument.
24704 See the *ctags* utility. The tags feature represented by **-t tagstring** and the **:t**
24705 command is optional. It shall be provided on any system that also provides a
24706 conforming implementation of *ctags*; otherwise, the use of **-t** produces undefined
24707 results.
- 24708 The filename resulting from the **-t** option shall be logically added as a prefix to the
24709 list of command line files, as if specified by the user. If the tag named by the
24710 *tagstring* argument is not found, it shall be an error, and *more* shall take no further
24711 action.
- 24712 If the tag specifies a line number, the first line of the display shall contain the
24713 beginning of that line. If the tag specifies a pattern, the first line of the display shall
24714 contain the beginning of the matching text from the first line of the file that
24715 contains that pattern. If the line does not exist in the file or matching text is not
24716 found, an informational message to this effect shall be displayed, and *more* shall
24717 display the default screen as if **-t** had not been specified.
- 24718 If both the **-t tagstring** and **-p command** options are given, the **-t tagstring** shall be
24719 processed first; that is, the file and starting line for the display shall be as specified
24720 by **-t**, and then the **-p more** command shall be executed. If the line (matching text)
24721 specified by the **-t** command does not exist (is not found), no **-p more** command
24722 shall be executed for this file at any time.
- 24723 **-u** Treat a **<backspace>** as a printable control character, displayed as an
24724 implementation-defined character sequence (see the EXTENDED DESCRIPTION
24725 section), suppressing backspacing and the special handling that produces
24726 underlined or standout mode text on some terminal types. Also, do not ignore a
24727 **<carriage-return>** at the end of a line.

24728 OPERANDS

24729 The following operand shall be supported:

- 24730 **file** A pathname of an input file. If no *file* operands are specified, the standard input
24731 shall be used. If a *file* is '**-**', the standard input shall be read at that point in the
24732 sequence.

24733 STDIN

24734 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '**-**'.

24735 INPUT FILES

24736 The input files being examined shall be text files. If standard output is a terminal, standard error
24737 shall be used to read commands from the user. If standard output is a terminal, standard error is
24738 not readable, and command input is needed, *more* may attempt to obtain user commands from
24739 the controlling terminal (for example, `/dev/tty`); otherwise, *more* shall terminate with an error
24740 indicating that it was unable to read user commands. If standard output is not a terminal, no
24741 error shall result if standard error cannot be opened for reading.

24742 ENVIRONMENT VARIABLES

24743 The following environment variables shall affect the execution of *more*:

24744 **COLUMNS** Override the system-selected horizontal display line size. See the Base Definitions
24745 volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables for valid values
24746 and results when it is unset or null.

24747 **EDITOR** Used by the **v** command to select an editor. See the EXTENDED DESCRIPTION
24748 section.

24749 **LANG** Provide a default value for the internationalization variables that are unset or null.
24750 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
24751 Internationalization Variables for the precedence of internationalization variables
24752 used to determine the values of locale categories.)

24753 **LC_ALL** If set to a non-empty string value, override the values of all the other
24754 internationalization variables.

24755 **LC_COLLATE**

24756 Determine the locale for the behavior of ranges, equivalence classes, and multi-
24757 character collating elements within regular expressions.

24758 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
24759 characters (for example, single-byte as opposed to multi-byte characters in
24760 arguments and input files) and the behavior of character classes within regular
24761 expressions.

24762 **LC_MESSAGES**

24763 Determine the locale that should be used to affect the format and contents of
24764 diagnostic messages written to standard error and informative messages written to
24765 standard output.

24766 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

24767 **LINES** Override the system-selected vertical screen size, used as the number of lines in a
24768 screenful. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8,
24769 Environment Variables for valid values and results when it is unset or null. The **-n**
24770 option shall take precedence over the **LINES** variable for determining the number
24771 of lines in a screenful.

24772 **MORE** Determine a string containing options described in the OPTIONS section preceded
24773 with hyphens and <blank>-separated as on the command line. Any command line
24774 options shall be processed after those in the **MORE** variable, as if the command
24775 line were:

24776 *more \$MORE options operands*

24777 The **MORE** variable shall take precedence over the **TERM** and **LINES** variables for
24778 determining the number of lines in a screenful.

TERM Determine the name of the terminal type. If this variable is unset or null, an unspecified default terminal type is used.

24781 ASYNCHRONOUS EVENTS

24782 Default.

24783 STDOUT

24784 The standard output shall be used to write the contents of the input files.

24785 STDERR

The standard error shall be used for diagnostic messages and user commands (see the INPUT FILES section), and, if standard output is a terminal device, to write a prompting string. The prompting string shall appear on the screen line below the last line of the file displayed in the current screenful. The prompt shall contain the name of the file currently being examined and shall contain an end-of-file indication and the name of the next file, if any, when prompting at the end-of-file. If an error or informational message is displayed, it is unspecified whether it is contained in the prompt. If it is not contained in the prompt, it shall be displayed and then the user shall be prompted for a continuation character, at which point another message or the user prompt may be displayed. The prompt is otherwise unspecified. It is unspecified whether informational messages are written for other user commands.

24796 OUTPUT FILES

24797 None.

24798 EXTENDED DESCRIPTION

The following section describes the behavior of *more* when the standard output is a terminal device. If the standard output is not a terminal device, no options other than **-s** shall have any effect, and all input files shall be copied to standard output otherwise unmodified, at which time *more* shall exit without further action.

The number of lines available per screen shall be determined by the **-n** option, if present, or by examining values in the environment (see the ENVIRONMENT VARIABLES section). If neither method yields a number, an unspecified number of lines shall be used.

The maximum number of lines written shall be one less than this number, because the screen line after the last line written shall be used to write a user prompt and user input. If the number of lines in the screen is less than two, the results are undefined. It is unspecified whether user input is permitted to be longer than the remainder of the single line where the prompt has been written.

24811 The number of columns available per line shall be determined by examining values in the
24812 environment (see the ENVIRONMENT VARIABLES section), with a default value as described
24813 in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

24814 Lines that are longer than the display shall be folded; the length at which folding occurs is
24815 unspecified, but should be appropriate for the output device. Folding may occur between glyphs
24816 of single characters that take up multiple display columns.

When standard output is a terminal and **-u** is not specified, **more** shall treat <backspace>s and <carriage-return>s specially:

- 24819 • A character, followed first by a sequence of n <backspace>s (where n is the same as the
24820 number of column positions that the character occupies), then by n underscore characters
24821 ('_'), shall cause that character to be written as underlined text, if the terminal type
24822 supports that. The n underscore characters, followed first by n <backspace>s, then any
24823 character with n column positions, shall also cause that character to be written as underlined
24824 text, if the terminal type supports that.

- 24825 • A sequence of n <backspace>s (where n is the same as the number of column positions that
24826 the previous character occupies) that appears between two identical printable characters
24827 shall cause the first of those two characters to be written as emboldened text (that is, visually
24828 brighter, standout mode, or inverse-video mode), if the terminal type supports that, and the
24829 second to be discarded. Immediately subsequent occurrences of <backspace>/character pairs
24830 for that same character shall also be discarded. (For example, the sequence "a\b\ba\b\ba\b" is
24831 interpreted as a single emboldened 'a'.)
- 24832 • The *more* utility shall logically discard all other <backspace>s from the line as well as the
24833 character which precedes them, if any.
- 24834 • A <carriage-return> at the end of a line shall be ignored, rather than being written as a non-
24835 printable character, as described in the next paragraph.

24836 It is implementation-defined how other non-printable characters are written. Implementations
24837 should use the same format that they use for the *ex print* command; see the OPTIONS section
24838 within the *ed* utility. It is unspecified whether a multi-column character shall be separated if it
24839 crosses a display line boundary; it shall not be discarded. The behavior is unspecified if the
24840 number of columns on the display is less than the number of columns any single character in the
24841 line being displayed would occupy.

24842 When each new file is displayed (or redisplayed), *more* shall write the first screen of the file.
24843 Once the initial screen has been written, *more* shall prompt for a user command. If the execution
24844 of the user command results in a screen that has lines in common with the current screen, and
24845 the device has sufficient terminal capabilities, *more* shall scroll the screen; otherwise, it is
24846 unspecified whether the screen is scrolled or redrawn.

24847 For all files but the last (including standard input if no file was specified, and for the last file as
24848 well, if the **-e** option was not specified), when *more* has written the last line in the file, *more* shall
24849 prompt for a user command. This prompt shall contain the name of the next file as well as an
24850 indication that *more* has reached end-of-file. If the user command is **f**, <control>-F, <space>, **j**,
24851 <newline>, **d**, <control>-D, or **s**, *more* shall display the next file. Otherwise, if displaying the last
24852 file, *more* shall exit. Otherwise, *more* shall execute the user command specified.

24853 Several of the commands described in this section display a previous screen from the input
24854 stream. In the case that text is being taken from a non-rewindable stream, such as a pipe, it is
24855 implementation-defined how much backwards motion is supported. If a command cannot be
24856 executed because of a limitation on backwards motion, an error message to this effect shall be
24857 displayed, the current screen shall not change, and the user shall be prompted for another
24858 command.

24859 If a command cannot be performed because there are insufficient lines to display, *more* shall alert
24860 the terminal. If a command cannot be performed because there are insufficient lines to display or
24861 a / command fails: if the input is the standard input, the last screen in the file may be displayed;
24862 otherwise, the current file and screen shall not change, and the user shall be prompted for another
24863 command.

24864 The interactive commands in the following sections shall be supported. Some commands can be
24865 preceded by a decimal integer, called *count* in the following descriptions. If not specified with
24866 the command, *count* shall default to 1. In the following descriptions, *pattern* is a basic regular
24867 expression, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3,
24868 Basic Regular Expressions. The term "examine" is historical usage meaning "open the file for
24869 viewing"; for example, *more foo* would be expressed as examining file *foo*.

24870 In the following descriptions, unless otherwise specified, *line* is a line in the *more* display, not a
24871 line from the file being examined.

24872 In the following descriptions, the *current position* refers to two things:

- 24873 1. The position of the current line on the screen
24874 2. The line number (in the file) of the current line on the screen

24875 Usually, the line on the screen corresponding to the current position is the third line on the
24876 screen. If this is not possible (there are fewer than three lines to display or this is the first page of
24877 the file, or it is the last page of the file), then the current position is either the first or last line on
24878 the screen as described later.

24879 **Help**

24880 *Synopsis:* h

24881 Write a summary of these commands and other implementation-defined commands. The
24882 behavior shall be as if the *more* utility were executed with the -e option on a file that contained
24883 the summary information. The user shall be prompted as described earlier in this section when
24884 end-of-file is reached. If the user command is one of those specified to continue to the next file,
24885 *more* shall return to the file and screen state from which the h command was executed.

24886 **Scroll Forward One Screenful**

24887 *Synopsis:* [count]f
24888 [count]<control>-F

24889 Scroll forward *count* lines, with a default of one screenful. If *count* is more than the screen size,
2490 only the final screenful shall be written.

24891 **Scroll Backward One Screenful**

24892 *Synopsis:* [count]b
24893 [count]<control>-B

24894 Scroll backward *count* lines, with a default of one screenful (see the -n option). If *count* is more
24895 than the screen size, only the final screenful shall be written.

24896 **Scroll Forward One Line**

24897 *Synopsis:* [count]<space>
24898 [count]j
24899 [count]<newline>

24900 Scroll forward *count* lines. The default *count* for the <space> shall be one screenful; for j and
24901 <newline>, one line. The entire *count* lines shall be written, even if *count* is more than the screen
24902 size.

24903 **Scroll Backward One Line**

24904 *Synopsis:* [count]k

24905 Scroll backward *count* lines. The entire *count* lines shall be written, even if *count* is more than the
24906 screen size.

24907 **Scroll Forward One Half Screenful**

24908 *Synopsis:* [count]d

24909 [count]<control>-D

24910 Scroll forward *count* lines, with a default of one half of the screen size. If *count* is specified, it
24911 shall become the new default for subsequent **d**, <control>-D, and **u** commands.

24912 **Skip Forward One Line**

24913 *Synopsis:* [count]s

24914 Display the screenful beginning with the line *count* lines after the last line on the current screen.
24915 If *count* would cause the current position to be such that less than one screenful would be
24916 written, the last screenful in the file shall be written.

24917 **Scroll Backward One Half Screenful**

24918 *Synopsis:* [count]u

24919 [count]<control>-U

24920 Scroll backward *count* lines, with a default of one half of the screen size. If *count* is specified, it
24921 shall become the new default for subsequent **d**, <control>-D, **u**, and <control>-U commands.
24922 The entire *count* lines shall be written, even if *count* is more than the screen size.

24923 **Go to Beginning of File**

24924 *Synopsis:* [count]g

24925 Display the screenful beginning with line *count*.

24926 **Go to End-of-File**

24927 *Synopsis:* [count]G

24928 If *count* is specified, display the screenful beginning with the line *count*. Otherwise, display the
24929 last screenful of the file.

24930 **Refresh the Screen**

24931 *Synopsis:* r

24932 <control>-L

24933 Refresh the screen.

24934 **Discard and Refresh**

24935 *Synopsis:* R

24936 Refresh the screen, discarding any buffered input. If the current file is non-seekable, buffered
24937 input shall not be discarded and the **R** command shall be equivalent to the **r** command.

24938 **Mark Position**

24939 *Synopsis:* `mletter`

24940 Mark the current position with the letter named by *letter*, where *letter* represents the name of one
24941 of the lowercase letters of the portable character set. When a new file is examined, all marks may
24942 be lost.

24943 **Return to Mark**

24944 *Synopsis:* `' letter`

24945 Return to the position that was previously marked with the letter named by *letter*, making that
24946 line the current position.

24947 **Return to Previous Position**

24948 *Synopsis:* `''`

24949 Return to the position from which the last large movement command was executed (where a
24950 “large movement” is defined as any movement of more than a screenful of lines). If no such
24951 movements have been made, return to the beginning of the file.

24952 **Search Forward for Pattern**

24953 *Synopsis:* `[count] /[!]pattern<newline>`

24954 Display the screenful beginning with the *count*th line containing the pattern. The search shall
24955 start after the first line currently displayed. The null regular expression ('/' followed by a
24956 <newline>) shall repeat the search using the previous regular expression, with a default *count*. If
24957 the character '!' is included, the matching lines shall be those that do not contain the *pattern*. If
24958 no match is found for the *pattern*, a message to that effect shall be displayed.

24959 **Search Backward for Pattern**

24960 *Synopsis:* `[count]?[!]pattern<newline>`

24961 Display the screenful beginning with the *count*th previous line containing the pattern. The search shall
24962 start on the last line before the first line currently displayed. The null regular
24963 expression ('?' followed by a <newline>) shall repeat the search using the previous regular
24964 expression, with a default *count*. If the character '!' is included, matching lines shall be those
24965 that do not contain the *pattern*. If no match is found for the *pattern*, a message to that effect shall
24966 be displayed.

24967 **Repeat Search**

24968 *Synopsis:* `[count]n`

24969 Repeat the previous search for *count*th line containing the last *pattern* (or not containing the last
24970 *pattern*, if the previous search was "/!" or "?!").

24971 **Repeat Search in Reverse**

24972 *Synopsis:* [*count*]N

24973 Repeat the search in the opposite direction of the previous search for the *count*th line containing
24974 the last *pattern* (or not containing the last *pattern*, if the previous search was "/!" or "?!").

24975 **Examine New File**

24976 *Synopsis:* :e [*filename*]<newline>

24977 Examine a new file. If the *filename* argument is not specified, the current file (see the :n and :p
24978 commands below) shall be re-examined. The *filename* shall be subjected to the process of shell
24979 word expansions (see Section 2.6 (on page 36)); if more than a single pathname results, the
24980 effects are unspecified. If *filename* is a number sign ('#'), the previously examined file shall be
24981 re-examined. If *filename* is not accessible for any reason (including that it is a non-seekable file),
24982 an error message to this effect shall be displayed and the current file and screen shall not change.

24983 **Examine Next File**

24984 *Synopsis:* [*count*]:n

24985 Examine the next file. If a number *count* is specified, the *count*th next file shall be examined. If
24986 *filename* refers to a non-seekable file, the results are unspecified.

24987 **Examine Previous File**

24988 *Synopsis:* [*count*]:p

24989 Examine the previous file. If a number *count* is specified, the *count*th previous file shall be
24990 examined. If *filename* refers to a non-seekable file, the results are unspecified.

24991 **Go to Tag**

24992 *Synopsis:* :t *tagstring*<newline>

24993 If the file containing the tag named by the *tagstring* argument is not the current file, examine the
24994 file, as if the :e command was executed with that file as the argument. Otherwise, or in addition,
24995 display the screenful beginning with the tag, as described for the -t option (see the OPTIONS
24996 section). If the ctags utility is not supported by the system, the use of :t produces undefined
24997 results.

24998 **Invoke Editor**

24999 *Synopsis:* v

25000 Invoke an editor to edit the current file being examined. If standard input is being examined, the
25001 results are unspecified. The name of the editor shall be taken from the environment variable
25002 EDITOR, or shall default to vi. If the last pathname component in EDITOR is either vi or ex, the
25003 editor shall be invoked with a -c *linenumber* command line argument, where *linenumber* is the
25004 line number of the file line containing the display line currently displayed as the first line of the
25005 screen. It is implementation-defined whether line-setting options are passed to editors other
25006 than vi and ex.

25007 When the editor exits, more shall resume with the same file and screen as when the editor was
25008 invoked.

25009 **Display Position**

25010 *Synopsis:* =

25011 <control>-G

25012 Write a message for which the information references the first byte of the line after the last line of

25013 the file on the screen. This message shall include the name of the file currently being examined,

25014 its number relative to the total number of files there are to examine, the line number in the file,

25015 the byte number and the total bytes in the file, and what percentage of the file precedes the

25016 current position. If *more* is reading from standard input, or the file is shorter than a single screen,

25017 the line number, the byte number, the total bytes, and the percentage need not be written.

25018 **Quit**

25019 *Synopsis:* q

25020 :q

25021 zz

25022 Exit *more*.

25023 **EXIT STATUS**

25024 The following exit values shall be returned:

25025 0 Successful completion.

25026 >0 An error occurred.

25027 **CONSEQUENCES OF ERRORS**

25028 If an error is encountered accessing a file when using the :n command, *more* shall attempt to

25029 examine the next file in the argument list, but the final exit status shall be affected. If an error is

25030 encountered accessing a file via the :p command, *more* shall attempt to examine the previous file

25031 in the argument list, but the final exit status shall be affected. If an error is encountered accessing

25032 a file via the :e command, *more* shall remain in the current file and the final exit status shall not

25033 be affected.

25034 **APPLICATION USAGE**

25035 When the standard output is not a terminal, only the -s filter-modification option is effective.

25036 This is based on historical practice. For example, a typical implementation of *man* pipes its

25037 output through *more* -s to squeeze excess white space for terminal users. When *man* is piped to

25038 *lp*, however, it is undesirable for this squeezing to happen.

25039 **EXAMPLES**

25040 The -p allows arbitrary commands to be executed at the start of each file. Examples are:

25041 *more -p G file1 file2*

25042 Examine each file starting with its last screenful.

25043 *more -p 100 file1 file2*

25044 Examine each file starting with line 100 in the current position (usually the third line, so line

25045 98 would be the first line written).

25046 *more -p /100 file1 file2*

25047 Examine each file starting with the first line containing the string "100" in the current

25048 position

25049 **RATIONALE**

25050 The *more* utility, available in BSD and BSD-derived systems, was chosen as the prototype for the

25051 POSIX file display program since it is more widely available than either the public-domain

25052 program *less* or than *pg*, a pager provided in System V. The 4.4 BSD *more* is the model for the

25053 features selected; it is almost fully upwards-compatible from the 4.3 BSD version in wide use
25054 and has become more amenable for *vi* users. Several features originally derived from various file
25055 editors, found in both *less* and *pg*, have been added to this volume of IEEE Std 1003.1-2001 as
25056 they have proved extremely popular with users.

25057 There are inconsistencies between *more* and *vi* that result from historical practice. For example,
25058 the single-character commands **h**, **f**, **b**, and <space> are screen movers in *more*, but cursor
25059 movers in *vi*. These inconsistencies were maintained because the cursor movements are not
25060 applicable to *more* and the powerful functionality achieved without the use of the control key
25061 justifies the differences.

25062 The tags interface has been included in a program that is not a text editor because it promotes
25063 another degree of consistent operation with *vi*. It is conceivable that the paging environment of
25064 *more* would be superior for browsing source code files in some circumstances.

25065 The operating mode referred to for block-mode terminals effectively adds a <newline> to each
25066 Synopsis line that currently has none. So, for example, **d<newline>** would page one screenful.
25067 The mode could be triggered by a command line option, environment variable, or some other
25068 method. The details are not imposed by this volume of IEEE Std 1003.1-2001 because there are so
25069 few systems known to support such terminals. Nevertheless, it was considered that all systems
25070 should be able to support *more* given the exception cited for this small community of terminals
25071 because, in comparison to *vi*, the cursor movements are few and the command set relatively
25072 amenable to the optional <newline>s.

25073 Some versions of *more* provide a shell escaping mechanism similar to the **ex !** command. The
25074 standard developers did not consider that this was necessary in a paginator, particularly given
25075 the wide acceptance of multiple window terminals and job control features. (They chose to
25076 retain such features in the editors and *mailx* because the shell interaction also gives an
25077 opportunity to modify the editing buffer, which is not applicable to *more*.)

25078 The **-p** (position) option replaces the **+** command because of the Utility Syntax Guidelines. In
25079 early proposals, it took a *pattern* argument, but historical *less* provided the *more* general facility of
25080 a command. It would have been desirable to use the same **-c** as *ex* and *vi*, but the letter was
25081 already in use.

25082 The text stating “from a non-rewindable stream … implementations may limit the amount of
25083 backwards motion supported” would allow an implementation that permitted no backwards
25084 motion beyond text already on the screen. It was not possible to require a minimum amount of
25085 backwards motion that would be effective for all conceivable device types. The implementation
25086 should allow the user to back up as far as possible, within device and reasonable memory
25087 allocation constraints.

25088 Historically, non-printable characters were displayed using the ARPA standard mappings,
25089 which are as follows:

- 25090 1. Printable characters are left alone.
- 25091 2. Control characters less than \177 are represented as followed by the character offset from
25092 the '@' character in the ASCII map; for example, \007 is represented as 'G'.
- 25093 3. \177 is represented as followed by '?'.

25094 The display of characters having their eighth bit set was less standard. Existing implementations
25095 use hex (0x00), octal (\000), and a meta-bit display. (The latter displayed characters with their
25096 eighth bit set as the two characters "M-", followed by the seven-bit display as described
25097 previously.) The latter probably has the best claim to historical practice because it was used with
25098 the **-v** option of 4 BSD and 4 BSD-derived versions of the *cat* utility since 1980.

25099 No specific display format is required by IEEE Std 1003.1-2001. Implementations are encouraged
25100 to conform to historic practice in the absence of any strong reason to diverge.

25101 FUTURE DIRECTIONS

25102 None.

25103 SEE ALSO

25104 Chapter 2 (on page 29), *ctags*, *ed*, *ex*, *vi*

25105 CHANGE HISTORY

25106 First released in Issue 4.

25107 Issue 5

25108 The FUTURE DIRECTIONS section is added.

25109 Issue 6

25110 This utility is marked as part of the User Portability Utilities option.

25111 The obsolescent SYNOPSIS is removed.

25112 The utility has been extensively reworked for alignment with the IEEE P1003.2b draft standard:

- Changes have been made as a result of IEEE PASC Interpretations 1003.2 #37 and #109.
- The *more* utility should be able to handle underlined and emboldened displays of characters that are wider than a single column position.

25116 NAME

25117 mv — move files

25118 SYNOPSIS

25119 mv [-fi] *source_file target_file*25120 mv [-fi] *source_file... target_file*

25121 DESCRIPTION

25122 In the first synopsis form, the *mv* utility shall move the file named by the *source_file* operand to the destination specified by the *target_file*. This first synopsis form is assumed when the final operand does not name an existing directory and is not a symbolic link referring to an existing directory.

25126 In the second synopsis form, *mv* shall move each file named by a *source_file* operand to a destination file in the existing directory named by the *target_dir* operand, or referenced if *target_dir* is a symbolic link referring to an existing directory. The destination path for each *source_file* shall be the concatenation of the target directory, a single slash character, and the last pathname component of the *source_file*. This second form is assumed when the final operand names an existing directory.

25132 If any operand specifies an existing file of a type not specified by the System Interfaces volume of IEEE Std 1003.1-2001, the behavior is implementation-defined.

25134 For each *source_file* the following steps shall be taken:

25135 1. If the destination path exists, the **-f** option is not specified, and either of the following conditions is true:

- 25137 a. The permissions of the destination path do not permit writing and the standard input
25138 is a terminal.
25139 b. The **-i** option is specified.

25140 the *mv* utility shall write a prompt to standard error and read a line from standard input. If
25141 the response is not affirmative, *mv* shall do nothing more with the current *source_file* and
25142 go on to any remaining *source_files*.

25143 2. The *mv* utility shall perform actions equivalent to the *rename()* function defined in the
25144 System Interfaces volume of IEEE Std 1003.1-2001, called with the following arguments:

- 25145 a. The *source_file* operand is used as the *old* argument.
25146 b. The destination path is used as the *new* argument.

25147 If this succeeds, *mv* shall do nothing more with the current *source_file* and go on to any
25148 remaining *source_files*. If this fails for any reasons other than those described for the *errno*
25149 [EXDEV] in the System Interfaces volume of IEEE Std 1003.1-2001, *mv* shall write a
25150 diagnostic message to standard error, do nothing more with the current *source_file*, and go
25151 on to any remaining *source_files*.

- 25152 3. If the destination path exists, and it is a file of type directory and *source_file* is not a file of
25153 type directory, or it is a file not of type directory and *source_file* is a file of type directory,
25154 *mv* shall write a diagnostic message to standard error, do nothing more with the current
25155 *source_file*, and go on to any remaining *source_files*.
25156 4. If the destination path exists, *mv* shall attempt to remove it. If this fails for any reason, *mv*
25157 shall write a diagnostic message to standard error, do nothing more with the current
25158 *source_file*, and go on to any remaining *source_files*.

25159 5. The file hierarchy rooted in *source_file* shall be duplicated as a file hierarchy rooted in the
25160 destination path. If *source_file* or any of the files below it in the hierarchy are symbolic
25161 links, the links themselves shall be duplicated, including their contents, rather than any
25162 files to which they refer. The following characteristics of each file in the file hierarchy shall
25163 be duplicated:

- 25164 • The time of last data modification and time of last access
25165 • The user ID and group ID
25166 • The file mode

25167 If the user ID, group ID, or file mode of a regular file cannot be duplicated, the file mode
25168 bits S_ISUID and S_ISGID shall not be duplicated.

25169 When files are duplicated to another file system, the implementation may require that the
25170 process invoking *mv* has read access to each file being duplicated.

25171 If the duplication of the file hierarchy fails for any reason, *mv* shall write a diagnostic
25172 message to standard error, do nothing more with the current *source_file*, and go on to any
25173 remaining *source_files*.

25174 If the duplication of the file characteristics fails for any reason, *mv* shall write a diagnostic
25175 message to standard error, but this failure shall not cause *mv* to modify its exit status.

25176 6. The file hierarchy rooted in *source_file* shall be removed. If this fails for any reason, *mv* shall
25177 write a diagnostic message to the standard error, do nothing more with the current
25178 *source_file*, and go on to any remaining *source_files*.

25179 OPTIONS

25180 The *mv* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
25181 Utility Syntax Guidelines.

25182 The following options shall be supported:

25183 **-f** Do not prompt for confirmation if the destination path exists. Any previous
25184 occurrence of the **-i** option is ignored.

25185 **-i** Prompt for confirmation if the destination path exists. Any previous occurrence of
25186 the **-f** option is ignored.

25187 Specifying more than one of the **-f** or **-i** options shall not be considered an error. The last option
25188 specified shall determine the behavior of *mv*.

25189 OPERANDS

25190 The following operands shall be supported:

25191 *source_file* A pathname of a file or directory to be moved.

25192 *target_file* A new pathname for the file or directory being moved.

25193 *target_dir* A pathname of an existing directory into which to move the input files.

25194 STDIN

25195 The standard input shall be used to read an input line in response to each prompt specified in
25196 the STDERR section. Otherwise, the standard input shall not be used.

25197 INPUT FILES

25198 The input files specified by each *source_file* operand can be of any file type.

25199 ENVIRONMENT VARIABLES

25200 The following environment variables shall affect the execution of *mv*:

25201 **LANG** Provide a default value for the internationalization variables that are unset or null.
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

25205 **LC_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

25207 LC_COLLATE

25208 Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the extended regular expression defined for
25209 the **yesexpr** locale keyword in the **LC_MESSAGES** category.

25211 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
25212 characters (for example, single-byte as opposed to multi-byte characters in
25213 arguments and input files), the behavior of character classes used in the extended
25214 regular expression defined for the **yesexpr** locale keyword in the **LC_MESSAGES**
25215 category.

25216 LC_MESSAGES

25217 Determine the locale for the processing of affirmative responses that should be
25218 used to affect the format and contents of diagnostic messages written to standard
25219 error.

25220 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

25221 ASYNCHRONOUS EVENTS

25222 Default.

25223 STDOUT

25224 Not used.

25225 STDERR

25226 Prompts shall be written to the standard error under the conditions specified in the
25227 DESCRIPTION section. The prompts shall contain the destination pathname, but their format is
25228 otherwise unspecified. Otherwise, the standard error shall be used only for diagnostic messages.

25229 OUTPUT FILES

25230 The output files may be of any file type.

25231 EXTENDED DESCRIPTION

25232 None.

25233 EXIT STATUS

25234 The following exit values shall be returned:

25235 0 All input files were moved successfully.

25236 >0 An error occurred.

25237 CONSEQUENCES OF ERRORS

25238 If the copying or removal of *source_file* is prematurely terminated by a signal or error, *mv* may
25239 leave a partial copy of *source_file* at the source or destination. The *mv* utility shall not modify
25240 both *source_file* and the destination path simultaneously; termination at any point shall leave
25241 either *source_file* or the destination path complete.

25242 APPLICATION USAGE

25243 Some implementations mark for update the *st_ctime* field of renamed files and some do not.
25244 Applications which make use of the *st_ctime* field may behave differently with respect to
25245 renamed files unless they are designed to allow for either behavior.

25246 EXAMPLES

25247 If the current directory contains only files **a** (of any type defined by the System Interfaces
25248 volume of IEEE Std 1003.1-2001), **b** (also of any type), and a directory **c**:

25249 mv a b c
25250 mv c d

25251 results with the original files **a** and **b** residing in the directory **d** in the current directory.

25252 RATIONALE

25253 Early proposals diverged from the SVID and BSD historical practice in that they required that
25254 when the destination path exists, the **-f** option is not specified, and input is not a terminal, *mv*
25255 fails. This was done for compatibility with *cp*. The current text returns to historical practice. It
25256 should be noted that this is consistent with the *rename()* function defined in the System
25257 Interfaces volume of IEEE Std 1003.1-2001, which does not require write permission on the
25258 target.

25259 For absolute clarity, paragraph (1), describing the behavior of *mv* when prompting for
25260 confirmation, should be interpreted in the following manner:

25261 if (exists AND (NOT f_option) AND
25262 ((not_writable AND input_is_terminal) OR i_option))

25263 The **-i** option exists on BSD systems, giving applications and users a way to avoid accidentally
25264 unlinking files when moving others. When the standard input is not a terminal, the 4.3 BSD *mv*
25265 deletes all existing destination paths without prompting, even when **-i** is specified; this is
25266 inconsistent with the behavior of the 4.3 BSD *cp* utility, which always generates an error when
25267 the file is unwritable and the standard input is not a terminal. The standard developers decided
25268 that use of **-i** is a request for interaction, so when the destination path exists, the utility takes
25269 instructions from whatever responds to standard input.

25270 The *rename()* function is able to move directories within the same file system. Some historical
25271 versions of *mv* have been able to move directories, but not to a different file system. The
25272 standard developers considered that this was an annoying inconsistency, so this volume of
25273 IEEE Std 1003.1-2001 requires directories to be able to be moved even across file systems. There
25274 is no **-R** option to confirm that moving a directory is actually intended, since such an option was
25275 not required for moving directories in historical practice. Requiring the application to specify it
25276 sometimes, depending on the destination, seemed just as inconsistent. The semantics of the
25277 *rename()* function were preserved as much as possible. For example, *mv* is not permitted to
25278 “rename” files to or from directories, even though they might be empty and removable.

25279 Historic implementations of *mv* did not exit with a non-zero exit status if they were unable to
25280 duplicate any file characteristics when moving a file across file systems, nor did they write a
25281 diagnostic message for the user. The former behavior has been preserved to prevent scripts from
25282 breaking; a diagnostic message is now required, however, so that users are alerted that the file
25283 characteristics have changed.

25284 The exact format of the interactive prompts is unspecified. Only the general nature of the
25285 contents of prompts are specified because implementations may desire more descriptive
25286 prompts than those used on historical implementations. Therefore, an application not using the
25287 **-f** option or using the **-i** option relies on the system to provide the most suitable dialog directly
25288 with the user, based on the behavior specified.

25289 When *mv* is dealing with a single file system and *source_file* is a symbolic link, the link itself is
25290 moved as a consequence of the dependence on the *rename()* functionality, per the
25291 DESCRIPTION. Across file systems, this has to be made explicit.

25292 **FUTURE DIRECTIONS**

25293 None.

25294 **SEE ALSO**

25295 *cp*, *In*, the System Interfaces volume of IEEE Std 1003.1-2001, *rename()*

25296 **CHANGE HISTORY**

25297 First released in Issue 2.

25298 **Issue 6**

25299 The *mv* utility is changed to describe processing of symbolic links as specified in the
25300 IEEE P1003.2b draft standard.

25301 The APPLICATION USAGE section is added.

25302 **NAME**

25303 newgrp — change to a new group

25304 **SYNOPSIS**

25305 UP newgrp [-l][group]

25306

25307 **DESCRIPTION**

25308 The *newgrp* utility shall create a new shell execution environment with a new real and effective
25309 group identification. Of the attributes listed in Section 2.12 (on page 61), the new shell execution
25310 environment shall retain the working directory, file creation mask, and exported variables from
25311 the previous environment (that is, open files, traps, unexported variables, alias definitions, shell
25312 functions, and *set* options may be lost). All other aspects of the process environment that are
25313 preserved by the *exec* family of functions defined in the System Interfaces volume of
25314 IEEE Std 1003.1-2001 shall also be preserved by *newgrp*; whether other aspects are preserved is
25315 unspecified.

25316 A failure to assign the new group identifications (for example, for security or password-related
25317 reasons) shall not prevent the new shell execution environment from being created.

25318 The *newgrp* utility shall affect the supplemental groups for the process as follows:

- 25319 • On systems where the effective group ID is normally in the supplementary group list (or
25320 whenever the old effective group ID actually is in the supplementary group list):

- 25321 — If the new effective group ID is also in the supplementary group list, *newgrp* shall change
25322 the effective group ID.

- 25323 — If the new effective group ID is not in the supplementary group list, *newgrp* shall add the
25324 new effective group ID to the list, if there is room to add it.

- 25325 • On systems where the effective group ID is not normally in the supplementary group list (or
25326 whenever the old effective group ID is not in the supplementary group list):

- 25327 — If the new effective group ID is in the supplementary group list, *newgrp* shall delete it.

- 25328 — If the old effective group ID is not in the supplementary list, *newgrp* shall add it if there is
25329 room.

25330 **Note:** The System Interfaces volume of IEEE Std 1003.1-2001 does not specify whether the effective
25331 group ID of a process is included in its supplementary group list.

25332 With no operands, *newgrp* shall change the effective group back to the groups identified in the
25333 user's user entry, and shall set the list of supplementary groups to that set in the user's group
25334 database entries.

25335 If a password is required for the specified group, and the user is not listed as a member of that
25336 group in the group database, the user shall be prompted to enter the correct password for that
25337 group. If the user is listed as a member of that group, no password shall be requested. If no
25338 password is required for the specified group, it is implementation-defined whether users not
25339 listed as members of that group can change to that group. Whether or not a password is
25340 required, implementation-defined system accounting or security mechanisms may impose
25341 additional authorization restrictions that may cause *newgrp* to write a diagnostic message and
25342 suppress the changing of the group identification.

25343 **OPTIONS**

25344 The *newgrp* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
25345 12.2, Utility Syntax Guidelines.

25346	The following option shall be supported:	
25347	-l	(The letter ell.) Change the environment to what would be expected if the user actually logged in again.
25349 OPERANDS		
25350	The following operand shall be supported:	
25351	<i>group</i>	A group name from the group database or a non-negative numeric group ID. Specifies the group ID to which the real and effective group IDs shall be set. If <i>group</i> is a non-negative numeric string and exists in the group database as a group name (see <i>getgrnam()</i>), the numeric group ID associated with that group name shall be used as the group ID.
25356 STDIN		
25357	Not used.	
25358 INPUT FILES		
25359	The file /dev/tty shall be used to read a single line of text for password checking, when one is required.	
25360		
25361 ENVIRONMENT VARIABLES		
25362	The following environment variables shall affect the execution of <i>newgrp</i> :	
25363	<i>LANG</i>	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
25364		
25365		
25366		
25367	<i>LC_ALL</i>	If set to a non-empty string value, override the values of all the other internationalization variables.
25368		
25369	<i>LC_CTYPE</i>	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
25370		
25371		
25372	<i>LC_MESSAGES</i>	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
25373		
25374		
25375 XSI	<i>NLSPATH</i>	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
25376 ASYNCHRONOUS EVENTS		
25377	Default.	
25378 STDOUT		
25379	Not used.	
25380 STDERR		
25381	The standard error shall be used for diagnostic messages and a prompt string for a password, if one is required. Diagnostic messages may be written in cases where the exit status is not available. See the EXIT STATUS section.	
25382		
25383		
25384 OUTPUT FILES		
25385	None.	
25386 EXTENDED DESCRIPTION		
25387	None.	

25388 EXIT STATUS

25389 If *newgrp* succeeds in creating a new shell execution environment, whether or not the group
25390 identification was changed successfully, the exit status shall be the exit status of the shell.
25391 Otherwise, the following exit value shall be returned:

25392 >0 An error occurred.

25393 CONSEQUENCES OF ERRORS

25394 The invoking shell may terminate.

25395 APPLICATION USAGE

25396 There is no convenient way to enter a password into the group database. Use of group
25397 passwords is not encouraged, because by their very nature they encourage poor security
25398 practices. Group passwords may disappear in the future.

25399 A common implementation of *newgrp* is that the current shell uses *exec* to overlay itself with
25400 *newgrp*, which in turn overlays itself with a new shell after changing group. On some
25401 implementations, however, this may not occur and *newgrp* may be invoked as a subprocess.

25402 The *newgrp* command is intended only for use from an interactive terminal. It does not offer a
25403 useful interface for the support of applications.

25404 The exit status of *newgrp* is generally inapplicable. If *newgrp* is used in a script, in most cases it
25405 successfully invokes a new shell and the rest of the original shell script is bypassed when the
25406 new shell exits. Used interactively, *newgrp* displays diagnostic messages to indicate problems.
25407 But usage such as:

```
25408 newgrp foo  
25409 echo $?
```

25410 is not useful because the new shell might not have access to any status *newgrp* may have
25411 generated (and most historical systems do not provide this status). A zero status echoed here
25412 does not necessarily indicate that the user has changed to the new group successfully. Following
25413 *newgrp* with the *id* command provides a portable means of determining whether the group
25414 change was successful or not.

25415 EXAMPLES

25416 None.

25417 RATIONALE

25418 Most historical implementations use one of the *exec* functions to implement the behavior of
25419 *newgrp*. Errors detected before the *exec* leave the environment unchanged, while errors detected
25420 after the *exec* leave the user in a changed environment. While it would be useful to have *newgrp*
25421 issue a diagnostic message to tell the user that the environment changed, it would be
25422 inappropriate to require this change to some historical implementations.

25423 The password mechanism is allowed in the group database, but how this would be
25424 implemented is not specified.

25425 The *newgrp* utility was retained in this volume of IEEE Std 1003.1-2001, even given the existence
25426 of the multiple group permissions feature in the System Interfaces volume of
25427 IEEE Std 1003.1-2001, for several reasons. First, in some implementations, the group ownership
25428 of a newly created file is determined by the group of the directory in which the file is created, as
25429 allowed by the System Interfaces volume of IEEE Std 1003.1-2001; on other implementations, the
25430 group ownership of a newly created file is determined by the effective group ID. On
25431 implementations of the latter type, *newgrp* allows files to be created with a specific group
25432 ownership. Finally, many implementations use the real group ID in accounting, and on such
25433 systems, *newgrp* allows the accounting identity of the user to be changed.

25434 FUTURE DIRECTIONS

25435 None.

25436 SEE ALSO

25437 Chapter 2 (on page 29), *sh*, the System Interfaces volume of IEEE Std 1003.1-2001, *exec*,
25438 *getgrnam()*

25439 CHANGE HISTORY

25440 First released in Issue 2.

25441 Issue 6

25442 This utility is marked as part of the User Portability Utilities option.

25443 The obsolescent SYNOPSIS is removed.

25444 The text describing supplemental groups is no longer conditional on {NGROUPS_MAX} being
25445 greater than 1. This is because {NGROUPS_MAX} now has a minimum value of 8. This is a FIPS
25446 requirement.

25447 NAME

25448 nice — invoke a utility with an altered nice value

25449 SYNOPSIS

25450 UP `nice [-n increment] utility [argument...]`

25451

25452 DESCRIPTION

25453 The *nice* utility shall invoke a utility, requesting that it be run with a different nice value (see the
25454 Base Definitions volume of IEEE Std 1003.1-2001, Section 3.239, Nice Value). With no options
25455 and only if the user has appropriate privileges, the executed utility shall be run with a nice value
25456 that is some implementation-defined quantity less than or equal to the nice value of the current
25457 process. If the user lacks appropriate privileges to affect the nice value in the requested manner,
25458 the *nice* utility shall not affect the nice value; in this case, a warning message may be written to
25459 standard error, but this shall not prevent the invocation of *utility* or affect the exit status.

25460 OPTIONS

25461 The *nice* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
25462 12.2, Utility Syntax Guidelines.

25463 The following option is supported:

25464 **-n increment** A positive or negative decimal integer which shall have the same effect on the
25465 execution of the utility as if the utility had called the *nice()* function with the
25466 numeric value of the *increment* option-argument.

25467 OPERANDS

25468 The following operands shall be supported:

25469 **utility** The name of a utility that is to be invoked. If the *utility* operand names any of the
25470 special built-in utilities in Section 2.14 (on page 64), the results are undefined.

25471 **argument** Any string to be supplied as an argument when invoking the utility named by the
25472 *utility* operand.

25473 STDIN

25474 Not used.

25475 INPUT FILES

25476 None.

25477 ENVIRONMENT VARIABLES

25478 The following environment variables shall affect the execution of *nice*:

25479 **LANG** Provide a default value for the internationalization variables that are unset or null.
25480 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,
25481 Internationalization Variables for the precedence of internationalization variables
25482 used to determine the values of locale categories.)

25483 **LC_ALL** If set to a non-empty string value, override the values of all the other
25484 internationalization variables.

25485 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
25486 characters (for example, single-byte as opposed to multi-byte characters in
25487 arguments).

25488 LC_MESSAGES

25489 Determine the locale that should be used to affect the format and contents of
25490 diagnostic messages written to standard error.

25491 XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
25492	PATH	Determine the search path used to locate the utility to be invoked. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.
25493		
25494 ASYNCHRONOUS EVENTS		
25495		Default.
25496 STDOUT		
25497		Not used.
25498 STDERR		
25499		The standard error shall be used only for diagnostic messages.
25500 OUTPUT FILES		
25501		None.
25502 EXTENDED DESCRIPTION		
25503		None.
25504 EXIT STATUS		
25505		If <i>utility</i> is invoked, the exit status of <i>nice</i> shall be the exit status of <i>utility</i> ; otherwise, the <i>nice</i> utility shall exit with one of the following values:
25506		
25507	1-125	An error occurred in the <i>nice</i> utility.
25508	126	The utility specified by <i>utility</i> was found but could not be invoked.
25509	127	The utility specified by <i>utility</i> could not be found.
25510 CONSEQUENCES OF ERRORS		
25511		Default.
25512 APPLICATION USAGE		
25513		The only guaranteed portable uses of this utility are:
25514	<i>nice utility</i>	
25515		Run <i>utility</i> with the default lower nice value.
25516	<i>nice -n <positive integer> utility</i>	
25517		Run <i>utility</i> with a lower nice value.
25518		On some implementations they have no discernible effect on the invoked utility and on some others they are exactly equivalent.
25519		
25520		Historical systems have frequently supported the <i><positive integer></i> up to 20. Since there is no
25521		error penalty associated with guessing a number that is too high, users without access to the
25522		system conformance document (to see what limits are actually in place) could use the historical
25523		1 to 20 range or attempt to use very large numbers if the job should be truly low priority.
25524		The nice value of a process can be displayed using the command:
25525	<code>ps -o nice</code>	
25526		The <i>command</i> , <i>env</i> , <i>nice</i> , <i>nohup</i> , <i>time</i> , and <i>xargs</i> utilities have been specified to use exit code 127 if
25527		an error occurs so that applications can distinguish “failure to find a utility” from “invoked
25528		utility exited with an error indication”. The value 127 was chosen because it is not commonly
25529		used for other meanings; most utilities use small values for “normal error conditions” and the
25530		values above 128 can be confused with termination due to receipt of a signal. The value 126 was
25531		chosen in a similar manner to indicate that the utility could be found, but not invoked. Some
25532		scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction
25533		between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to

25534 exec the utility fail with [ENOENT], and uses 126 when any attempt to exec the utility fails for
25535 any other reason.

25536 **EXAMPLES**

25537 None.

25538 **RATIONALE**

25539 Due to the text about the limits of the nice value being implementation-defined, *nice* is not
25540 actually required to change the nice value of the executed command; the limits could be zero
25541 differences from the system default, although the implementor is required to document this fact
25542 in the conformance document.

25543 The 4.3 BSD version of *nice* does not check whether *increment* is a valid decimal integer. The
25544 command *nice -x utility*, for example, would be treated the same as the command *nice --1 utility*. If the user does not have appropriate privileges, this results in a “permission denied”
25546 error. This is considered a bug.

25547 When a user without appropriate privileges gives a negative *increment*, System V treats it like
25548 the command *nice -0 utility*, while 4.3 BSD writes a “permission denied” message and does not
25549 run the utility. Neither was considered clearly superior, so the behavior was left unspecified.

25550 The C shell has a built-in version of *nice* that has a different interface from the one described in
25551 this volume of IEEE Std 1003.1-2001.

25552 The term “utility” is used, rather than “command”, to highlight the fact that shell compound
25553 commands, pipelines, and so on, cannot be used. Special built-ins also cannot be used.
25554 However, “utility” includes user application programs and shell scripts, not just utilities defined
25555 in this volume of IEEE Std 1003.1-2001.

25556 Historical implementations of *nice* provide a nice value range of 40 or 41 discrete steps, with the
25557 default nice value being the midpoint of that range. By default, they lower the nice value of the
25558 executed utility by 10.

25559 Some historical documentation states that the *increment* value must be within a fixed range. This
25560 is misleading; the valid *increment* values on any invocation are determined by the current
25561 process nice value, which is not always the default.

25562 The definition of nice value is not intended to suggest that all processes in a system have
25563 priorities that are comparable. Scheduling policy extensions such as the realtime priorities in the
25564 System Interfaces volume of IEEE Std 1003.1-2001 make the notion of a single underlying
25565 priority for all scheduling policies problematic. Some implementations may implement the *nice*-
25566 related features to affect all processes on the system, others to affect just the general time-
25567 sharing activities implied by this volume of IEEE Std 1003.1-2001, and others may have no effect
25568 at all. Because of the use of “implementation-defined” in *nice* and *renice*, a wide range of
25569 implementation strategies are possible.

25570 **FUTURE DIRECTIONS**

25571 None.

25572 **SEE ALSO**

25573 Chapter 2 (on page 29), *renice*, the System Interfaces volume of IEEE Std 1003.1-2001, *nice()*

25574 **CHANGE HISTORY**

25575 First released in Issue 4.

25576 Issue 6

25577 This utility is marked as part of the User Portability Utilities option.

25578 The obsolescent SYNOPSIS is removed.

25579 NAME

25580 nl — line numbering filter

25581 SYNOPSIS

```
25582 XSI    nl [-p][-b type][-d delim][-f type][-h type][-i incr][-l num][-n format]
25583          [-s sep][-v startnum][-w width][file]
```

25584

25585 DESCRIPTION

25586 The *nl* utility shall read lines from the named *file* or the standard input if no *file* is named and
 25587 shall reproduce the lines to standard output. Lines shall be numbered on the left. Additional
 25588 functionality may be provided in accordance with the command options in effect.

25589 The *nl* utility views the text it reads in terms of logical pages. Line numbering shall be reset at
 25590 the start of each logical page. A logical page consists of a header, a body, and a footer section.
 25591 Empty sections are valid. Different line numbering options are independently available for
 25592 header, body, and footer (for example, no numbering of header and footer lines while
 25593 numbering blank lines only in the body).

25594 The starts of logical page sections shall be signaled by input lines containing nothing but the
 25595 following delimiter characters:

25596
25597
25598
25599

Line	Start of
\:\:\:\:	Header
\:\:\:	Body
\:	Footer

25600 Unless otherwise specified, *nl* shall assume the text being read is in a single logical page body.

25601 OPTIONS

25602 The *nl* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,
 25603 Utility Syntax Guidelines. Only one file can be named.

25604 The following options shall be supported:

25605 **-b type** Specify which logical page body lines shall be numbered. Recognized *types* and
 25606 their meaning are:

25607 **a** Number all lines.

25608 **t** Number only non-empty lines.

25609 **n** No line numbering.

25610 **pstring** Number only lines that contain the basic regular expression specified in
 25611 *string*.

25612 The default *type* for logical page body shall be **t** (text lines numbered).

25613 **-d delim** Specify the delimiter characters that indicate the start of a logical page section.
 25614 These can be changed from the default characters "\:" to two user-specified
 25615 characters. If only one character is entered, the second character shall remain the
 25616 default character ':'.

25617 **-f type** Specify the same as **b type** except for footer. The default for logical page footer
 25618 shall be **n** (no lines numbered).

25619 **-h type** Specify the same as **b type** except for header. The default *type* for logical page
 25620 header shall be **n** (no lines numbered).

25621	-i incr	Specify the increment value used to number logical page lines. The default shall be 1.
25623	-l num	Specify the number of blank lines to be considered as one. For example, -l 2 results in only the second adjacent blank line being numbered (if the appropriate -h a , -b a , or -f a option is set). The default shall be 1.
25626	-n format	Specify the line numbering format. Recognized values are: ln , left justified, leading zeros suppressed; rn , right justified, leading zeros suppressed; rz , right justified, leading zeros kept. The default <i>format</i> shall be rn (right justified).
25629	-p	Specify that numbering should not be restarted at logical page delimiters.
25630	-s sep	Specify the characters used in separating the line number and the corresponding text line. The default <i>sep</i> shall be a <tab>.
25632	-v startnum	Specify the initial value used to number logical page lines. The default shall be 1.
25633	-w width	Specify the number of characters to be used for the line number. The default <i>width</i> shall be 6.
25634		

25635 OPERANDS

25636 The following operand shall be supported:

25637	<i>file</i>	A pathname of a text file to be line-numbered.
-------	-------------	--

25638 STDIN

25639 The standard input is a text file that is used if no *file* operand is given.

25640 INPUT FILES

25641 The input file named by the *file* operand is a text file.

25642 ENVIRONMENT VARIABLES

25643 The following environment variables shall affect the execution of *nl*:

25644	LANG	Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
25648	LC_ALL	If set to a non-empty string value, override the values of all the other internationalization variables.
25649		
25650	LC_COLLATE	Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.
25653	LC_CTYPE	Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), the behavior of character classes within regular expressions, and for deciding which characters are in character class graph (for the -b t , -f t , and -h t options).
25658	LC_MESSAGES	Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
25659		
25660		
25661	NLSPATH	Determine the location of message catalogs for the processing of LC_MESSAGES .

25662 ASYNCHRONOUS EVENTS

25663 Default.

25664 STDOUT

25665 The standard output shall be a text file in the following format:

25666 "%s%s%s", <line number>, <separator>, <input line>

25667 where <line number> is one of the following numeric formats:

25668 %6d When the **rn** format is used (the default; see **-n**).

25669 %06d When the **rz** format is used.

25670 %-6d When the **ln** format is used.

25671 <empty> When line numbers are suppressed for a portion of the page; the <separator> is also suppressed.

25673 In the preceding list, the number 6 is the default width; the **-w** option can change this value.

25674 STDERR

25675 The standard error shall be used only for diagnostic messages.

25676 OUTPUT FILES

25677 None.

25678 EXTENDED DESCRIPTION

25679 None.

25680 EXIT STATUS

25681 The following exit values shall be returned:

25682 0 Successful completion.

25683 >0 An error occurred.

25684 CONSEQUENCES OF ERRORS

25685 Default.

25686 APPLICATION USAGE

25687 In using the **-d** *delim* option, care should be taken to escape characters that have special meaning to the command interpreter.

25689 EXAMPLES

25690 The command:

25691 nl -v 10 -i 10 -d \!+ file1

25692 numbers *file1* starting at line number 10 with an increment of 10. The logical page delimiter is " !+". Note that the ' !' has to be escaped when using *csh* as a command interpreter because of 25693 its history substitution syntax. For *ksh* and *sh* the escape is not necessary, but does not do any 25694 harm.

25696 RATIONALE

25697 None.

25698 FUTURE DIRECTIONS

25699 None.

25700 SEE ALSO

25701 *pr*

25702 CHANGE HISTORY

25703 First released in Issue 2.

25704 Issue 5

25705 The option [-f *type*] is added to the SYNOPSIS. The option descriptions are presented in
25706 alphabetic order. The description of -bt is changed to “Number only non-empty lines”.

25707 Issue 6

25708 The obsolescent behavior allowing the options to be intermingled with the optional *file* operand
25709 is removed.

25710 NAME

25711 nm — write the name list of an object file (**DEVELOPMENT**)

25712 SYNOPSIS

25713 UP SD XSI nm [-APv][-efox][-g|-u][-t *format*] *file...*

25714

25715 DESCRIPTION

25716 This utility shall be provided on systems that support both the User Portability Utilities option
25717 and the Software Development Utilities option. On other systems it is optional. Certain options
25718 are only available on XSI-conformant systems.

25719 The *nm* utility shall display symbolic information appearing in the object file, executable file, or
25720 object-file library named by *file*. If no symbolic information is available for a valid input file, the
25721 *nm* utility shall report that fact, but not consider it an error condition.

25722 XSI The default base used when numeric values are written is unspecified. On XSI-conformant
25723 systems, it shall be decimal.

25724 OPTIONS

25725 The *nm* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section
25726 12.2, Utility Syntax Guidelines.

25727 The following options shall be supported:

25728 **-A** Write the full pathname or library name of an object on each line.

25729 XSI **-e** Write only external (global) and static symbol information.

25730 XSI **-f** Produce full output. Write redundant symbols (.text, .data, and .bss), normally
25731 suppressed.

25732 **-g** Write only external (global) symbol information.

25733 XSI **-o** Write numeric values in octal (equivalent to **-t o**).

25734 **-P** Write information in a portable output format, as specified in the STDOUT section.

25735 **-t *format*** Write each numeric value in the specified format. The format shall be dependent
25736 on the single character used as the *format* option-argument:

25737 XSI **d** The offset is written in decimal (default).

25738 **o** The offset is written in octal.

25739 **x** The offset is written in hexadecimal.

25740 **-u** Write only undefined symbols.

25741 **-v** Sort output by value instead of alphabetically.

25742 XSI **-x** Write numeric values in hexadecimal (equivalent to **-t x**).

25743 OPERANDS

25744 The following operand shall be supported:

25745 *file* A pathname of an object file, executable file, or object-file library.

25746 STDIN

25747 See the INPUT FILES section.

25748 INPUT FILES

25749 The input file shall be an object file, an object-file library whose format is the same as those produced by the *ar* utility for link editing, or an executable file. The *nm* utility may accept additional implementation-defined object library formats for the input file.

25752 ENVIRONMENT VARIABLES

25753 The following environment variables shall affect the execution of *nm*:

25754 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

25758 *LC_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

LC_COLLATE

25761 Determine the locale for character collation information for the symbol-name and symbol-value collation sequences.

25763 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

LC_MESSAGES

25767 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

25769 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

25770 ASYNCHRONOUS EVENTS

25771 Default.

25772 STDOUT

25773 If symbolic information is present in the input files, then for each file or for each member of an archive, the *nm* utility shall write the following information to standard output. By default, the format is unspecified, but the output shall be sorted alphabetically by symbol name:

- Library or object name, if *-A* is specified

- Symbol name

- Symbol type, which shall either be one of the following single characters or an implementation-defined type represented by a single character:

- A Global absolute symbol.

- a Local absolute symbol.

- B Global “bss” (that is, uninitialized data space) symbol.

- b Local bss symbol.

- D Global data symbol.

- d Local data symbol.

- T Global text symbol.

- t Local text symbol.

- U Undefined symbol.

- 25789 • Value of the symbol
25790 • The size associated with the symbol, if applicable
- 25791 This information may be supplemented by additional information specific to the
25792 implementation.
- 25793 If the **-P** option is specified, the previous information shall be displayed using the following
25794 portable format. The three versions differ depending on whether **-t d**, **-t o**, or **-t x** was specified,
25795 respectively:
- ```
25796 "%s%s %s %d\n", <library/object name>, <name>, <type>,
25797 <value>, <size>
25798 "%s%s %s %o\n", <library/object name>, <name>, <type>,
25799 <value>, <size>
25800 "%s%s %s %x\n", <library/object name>, <name>, <type>,
25801 <value>, <size>
```
- 25802 where *<library/object name>* shall be formatted as follows:
- 25803   • If **-A** is not specified, *<library/object name>* shall be an empty string.
  - 25804   • If **-A** is specified and the corresponding *file* operand does not name a library:  
25805        "*%s*: ", <file>
  - 25806   • If **-A** is specified and the corresponding *file* operand names a library. In this case, *<object file>*  
25807        shall name the object file in the library containing the symbol being described:  
25808        "*%s[%s]*: ", <file>, <object file>
- 25809 If **-A** is not specified, then if more than one *file* operand is specified or if only one *file* operand is  
25810 specified and it names a library, *nm* shall write a line identifying the object containing the  
25811 following symbols before the lines containing those symbols, in the form:
- 25812   • If the corresponding *file* operand does not name a library:  
25813        "*%s:\n*", <file>
  - 25814   • If the corresponding *file* operand names a library; in this case, *<object file>* shall be the name  
25815        of the file in the library containing the following symbols:  
25816        "*%s[%s]:\n*", <file>, <object file>
- 25817 If **-P** is specified, but **-t** is not, the format shall be as if **-t x** had been specified.
- 25818 **STDERR**  
25819       The standard error shall be used only for diagnostic messages.
- 25820 **OUTPUT FILES**  
25821       None.
- 25822 **EXTENDED DESCRIPTION**  
25823       None.
- 25824 **EXIT STATUS**  
25825       The following exit values shall be returned:
- 25826       0   Successful completion.
  - 25827      >0   An error occurred.

25828 **CONSEQUENCES OF ERRORS**

25829 Default.

25830 **APPLICATION USAGE**

25831 Mechanisms for dynamic linking make this utility less meaningful when applied to an executable file because a dynamically linked executable may omit numerous library routines that would be found in a statically linked executable.

25834 **EXAMPLES**

25835 None.

25836 **RATIONALE**

25837 Historical implementations of *nm* have used different bases for numeric output and supplied different default types of symbols that were reported. The **-t** *format* option, similar to that used in *od* and *strings*, can be used to specify the numeric base; **-g** and **-u** can be used to restrict the amount of output or the types of symbols included in the output.

25841 The compromise of using **-t** *format* versus using **-d**, **-o**, and other similar options was necessary because of differences in the meaning of **-o** between implementations. The **-o** option from BSD has been provided here as **-A** to avoid confusion with the **-o** from System V (which has been provided here as **-t** and as **-o** on XSI-conformant systems).

25845 The option list was significantly reduced from that provided by historical implementations.

25846 The *nm* description is a subset of both the System V and BSD *nm* utilities with no specified default output.

25848 It was recognized that mechanisms for dynamic linking make this utility less meaningful when applied to an executable file (because a dynamically linked executable file may omit numerous library routines that would be found in a statically linked executable file), but the value of *nm* during software development was judged to outweigh other limitations.

25852 The default output format of *nm* is not specified because of differences in historical implementations. The **-P** option was added to allow some type of portable output format. After a comparison of the different formats used in SunOS, BSD, SVR3, and SVR4, it was decided to create one that did not match the current format of any of these four systems. The format devised is easy to parse by humans, easy to parse in shell scripts, and does not need to vary depending on locale (because no English descriptions are included). All of the systems currently have the information available to use this format.

25859 The format given in *nm* STDOUT uses spaces between the fields, which may be any number of <blank>s required to align the columns. The single-character types were selected to match historical practice, and the requirement that implementation additions also be single characters made parsing the information easier for shell scripts.

25863 **FUTURE DIRECTIONS**

25864 None.

25865 **SEE ALSO**

25866 *ar*, *c99*

25867 **CHANGE HISTORY**

25868 First released in Issue 2.

25869 **Issue 6**

25870 This utility is marked as supported when both the User Portability Utilities option and the Software Development Utilities option are supported.

**25872 NAME**

25873        nohup — invoke a utility immune to hangups

**25874 SYNOPSIS**

25875        nohup *utility* [*argument...*]

**25876 DESCRIPTION**

25877        The *nohup* utility shall invoke the utility named by the *utility* operand with arguments supplied  
25878        as the *argument* operands. At the time the named *utility* is invoked, the SIGHUP signal shall be  
25879        set to be ignored.

25880        If the standard output is a terminal, all output written by the named *utility* to its standard output  
25881        shall be appended to the end of the file **nohup.out** in the current directory. If **nohup.out** cannot  
25882        be created or opened for appending, the output shall be appended to the end of the file  
25883        **nohup.out** in the directory specified by the *HOME* environment variable. If neither file can be  
25884        created or opened for appending, *utility* shall not be invoked. If a file is created, the file's  
25885        permission bits shall be set to S\_IRUSR | S\_IWUSR.

25886        If the standard error is a terminal, all output written by the named *utility* to its standard error  
25887        shall be redirected to the same file descriptor as the standard output.

**25888 OPTIONS**

25889        None.

**25890 OPERANDS**

25891        The following operands shall be supported:

25892        *utility*        The name of a utility that is to be invoked. If the *utility* operand names any of the  
25893        special built-in utilities in Section 2.14 (on page 64), the results are undefined.

25894        *argument*     Any string to be supplied as an argument when invoking the utility named by the  
25895        *utility* operand.

**25896 STDIN**

25897        Not used.

**25898 INPUT FILES**

25899        None.

**25900 ENVIRONMENT VARIABLES**

25901        The following environment variables shall affect the execution of *nohup*:

25902        *HOME*        Determine the pathname of the user's home directory: if the output file **nohup.out**  
25903        cannot be created in the current directory, the *nohup* utility shall use the directory  
25904        named by *HOME* to create the file.

25905        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
25906        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
25907        Internationalization Variables for the precedence of internationalization variables  
25908        used to determine the values of locale categories.)

25909        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
25910        internationalization variables.

25911        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
25912        characters (for example, single-byte as opposed to multi-byte characters in  
25913        arguments).

**25914 LC\_MESSAGES**

25915        Determine the locale that should be used to affect the format and contents of

|           |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25916     |                               | diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 25917 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 25918     | <b>PATH</b>                   | Determine the search path that is used to locate the utility to be invoked. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 25919     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25920     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25921     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25922     |                               | The <i>nohup</i> utility shall take the standard action for all signals except that SIGHUP shall be ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 25923     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25924     | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25925     |                               | If the standard output is not a terminal, the standard output of <i>nohup</i> shall be the standard output generated by the execution of the <i>utility</i> specified by the operands. Otherwise, nothing shall be written to the standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 25926     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25927     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25928     | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25929     |                               | If the standard output is a terminal, a message shall be written to the standard error, indicating the name of the file to which the output is being appended. The name of the file shall be either <b>nohup.out</b> or <b>\$HOME/nohup.out</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25930     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25931     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25932     | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25933     |                               | If the standard output is a terminal, all output written by the named <i>utility</i> to the standard output and standard error is appended to the file <b>nohup.out</b> , which is created if it does not already exist.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 25934     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25935     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25936     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25937     |                               | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 25938     | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25939     |                               | The following exit values shall be returned:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 25940     | 126                           | The utility specified by <i>utility</i> was found but could not be invoked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 25941     | 127                           | An error occurred in the <i>nohup</i> utility or the utility specified by <i>utility</i> could not be found.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 25942     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25943     |                               | Otherwise, the exit status of <i>nohup</i> shall be that of the utility specified by the <i>utility</i> operand.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 25944     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25945     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 25946     | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25947     |                               | The <i>command</i> , <i>env</i> , <i>nice</i> , <i>nohup</i> , <i>time</i> , and <i>xargs</i> utilities have been specified to use exit code 127 if an error occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited with an error indication”. The value 127 was chosen because it is not commonly used for other meanings; most utilities use small values for “normal error conditions” and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked. Some scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to <i>exec</i> the utility fail with [ENOENT], and uses 126 when any attempt to <i>exec</i> the utility fails for any other reason. |
| 25948     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25949     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25950     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25951     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25952     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25953     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25954     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25955     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25956     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25957     | <b>EXAMPLES</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25958     |                               | It is frequently desirable to apply <i>nohup</i> to pipelines or lists of commands. This can be done by placing pipelines and command lists in a single file; this file can then be invoked as a utility, and the <i>nohup</i> applies to everything in the file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25959     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25960     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

25961        Alternatively, the following command can be used to apply *nohup* to a complex command:

25962        nohup sh -c '*complex-command-line*'

25963 **RATIONALE**

25964        The 4.3 BSD version ignores SIGTERM and SIGHUP, and if *./nohup.out* cannot be used, it fails  
25965 instead of trying to use \$HOME/nohup.out.

25966        The *csh* utility has a built-in version of *nohup* that acts differently from the *nohup* defined in this  
25967 volume of IEEE Std 1003.1-2001.

25968        The term *utility* is used, rather than *command*, to highlight the fact that shell compound  
25969 commands, pipelines, special built-ins, and so on, cannot be used directly. However, *utility*  
25970 includes user application programs and shell scripts, not just the standard utilities.

25971        Historical versions of the *nohup* utility use default file creation semantics. Some more recent  
25972 versions use the permissions specified here as an added security precaution.

25973        Some historical implementations ignore SIGQUIT in addition to SIGHUP; others ignore  
25974 SIGTERM. An early proposal allowed, but did not require, SIGQUIT to be ignored. Several  
25975 reviewers objected that *nohup* should only modify the handling of SIGHUP as required by this  
25976 volume of IEEE Std 1003.1-2001.

25977 **FUTURE DIRECTIONS**

25978        None.

25979 **SEE ALSO**

25980        Chapter 2 (on page 29), *sh*, the System Interfaces volume of IEEE Std 1003.1-2001, *signal()*

25981 **CHANGE HISTORY**

25982        First released in Issue 2.

## 25983 NAME

25984 od — dump files in various formats

## 25985 SYNOPSIS

25986 od [-v][-A address\_base][-j skip][-N count][-t type\_string]...  
25987 [file...]

25988 XSI od [-bcdosx][file] [+][offset[.][b]]

25989

## 25990 DESCRIPTION

25991 The *od* utility shall write the contents of its input files to standard output in a user-specified  
25992 format.

## 25993 OPTIONS

25994 The *od* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
25995 XSI Utility Syntax Guidelines, except that the order of presentation of the **-t** options and the  
25996 **-bcdosx** options is significant.

25997 The following options shall be supported:

25998 **-A address\_base**25999 Specify the input offset base. See the EXTENDED DESCRIPTION section. The  
26000 application shall ensure that the *address\_base* option-argument is a character. The  
26001 characters 'd', 'o', and 'x' specify that the offset base shall be written in  
26002 decimal, octal, or hexadecimal, respectively. The character 'n' specifies that the  
26003 offset shall not be written.26004 XSI **-b** Interpret bytes in octal. This shall be equivalent to **-t o1**.26005 XSI **-c** Interpret bytes as characters specified by the current setting of the *LC\_CTYPE*  
26006 category. Certain non-graphic characters appear as C escapes: "NUL=\0",  
26007 "BS=\b", "FF=\f", "NL=\n", "CR=\r", "HT=\t"; others appear as 3-digit octal  
26008 numbers.26009 XSI **-d** Interpret words (two-byte units) in unsigned decimal. This shall be equivalent to  
26010 **-t u2**.26011 **-j skip** Jump over *skip* bytes from the beginning of the input. The *od* utility shall read or  
26012 seek past the first *skip* bytes in the concatenated input files. If the combined input  
26013 is not at least *skip* bytes long, the *od* utility shall write a diagnostic message to  
26014 standard error and exit with a non-zero exit status.26015 By default, the *skip* option-argument shall be interpreted as a decimal number.  
26016 With a leading 0x or 0X, the offset shall be interpreted as a hexadecimal number;  
26017 otherwise, with a leading '0', the offset shall be interpreted as an octal number.  
26018 Appending the character 'b', 'k', or 'm' to offset shall cause it to be interpreted  
26019 as a multiple of 512, 1 024, or 1 048 576 bytes, respectively. If the *skip* number is  
26020 hexadecimal, any appended 'b' shall be considered to be the final hexadecimal  
26021 digit.26022 **-N count** Format no more than *count* bytes of input. By default, *count* shall be interpreted as  
26023 a decimal number. With a leading 0x or 0X, *count* shall be interpreted as a  
26024 hexadecimal number; otherwise, with a leading '0', it shall be interpreted as an  
26025 octal number. If *count* bytes of input (after successfully skipping, if **-j skip** is  
26026 specified) are not available, it shall not be considered an error; the *od* utility shall  
26027 format the input that is available.

|           |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26028 XSI | <b>-o</b>                                                                                                    | Interpret <i>words</i> (two-byte units) in octal. This shall be equivalent to <b>-t o2</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 26029 XSI | <b>-s</b>                                                                                                    | Interpret <i>words</i> (two-byte units) in signed decimal. This shall be equivalent to <b>-t d2</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 26030     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26031     | <b>-t type_string</b>                                                                                        | Specify one or more output types. See the EXTENDED DESCRIPTION section. The application shall ensure that the <i>type_string</i> option-argument is a string specifying the types to be used when writing the input data. The string shall consist of the type specification characters a, c, d, f, o, u, and x, specifying named character, character, signed decimal, floating point, octal, unsigned decimal, and hexadecimal, respectively. The type specification characters d, f, o, u, and x can be followed by an optional unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type. The type specification character f can be followed by an optional F, D, or L indicating that the conversion should be applied to an item of type <b>float</b> , <b>double</b> , or <b>long double</b> , respectively. The type specification characters d, o, u, and x can be followed by an optional C, S, I, or L indicating that the conversion should be applied to an item of type <b>char</b> , <b>short</b> , <b>int</b> , or <b>long</b> , respectively. Multiple types can be concatenated within the same <i>type_string</i> and multiple <b>-t</b> options can be specified. Output lines shall be written for each type specified in the order in which the type specification characters are specified. |
| 26032     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26033     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26034     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26035     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26036     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26037     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26038     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26039     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26040     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26041     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26042     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26043     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26044     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26045     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26046     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26047     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26048     | <b>-v</b>                                                                                                    | Write all input data. Without the <b>-v</b> option, any number of groups of output lines, which would be identical to the immediately preceding group of output lines (except for the byte offsets), shall be replaced with a line containing only an asterisk ('*').                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 26049     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26050     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26051     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26052 XSI | <b>-x</b>                                                                                                    | Interpret <i>words</i> (two-byte units) in hexadecimal. This shall be equivalent to <b>-t x2</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 26053 XSI |                                                                                                              | Multiple types can be specified by using multiple <b>-bcdostx</b> options. Output lines are written for each type specified in the order in which the types are specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 26054     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26055     | <b>OPERANDS</b>                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26056     | The following operands shall be supported:                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26057     | <i>file</i>                                                                                                  | A pathname of a file to be read. If no <i>file</i> operands are specified, the standard input shall be used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 26058     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26059     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26060     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26061     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26062 XSI |                                                                                                              | If there are no more than two operands, none of the <b>-A</b> , <b>-j</b> , <b>-N</b> , or <b>-t</b> options is specified, and either of the following is true: the first character of the last operand is a plus sign ('+'), or there are two operands and the first character of the last operand is numeric; the last operand shall be interpreted as an offset operand on XSI-conformant systems. Under these conditions, the results are unspecified on systems that are not XSI-conformant systems.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26063     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26064     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26065 XSI | <b>[+]offset[.][]b</b>                                                                                       | The <i>offset</i> operand specifies the offset in the file where dumping is to commence. This operand is normally interpreted as octal bytes. If '.' is appended, the offset shall be interpreted in decimal. If 'b' is appended, the offset shall be interpreted in units of 512 bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26066     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26067     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26068     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26069     | <b>STDIN</b>                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26070     | The standard input shall be used only if no <i>file</i> operands are specified. See the INPUT FILES section. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26071     |                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

**26072 INPUT FILES**

26073 The input files can be any file type.

**26074 ENVIRONMENT VARIABLES**

26075 The following environment variables shall affect the execution of *od*:

26076 **LANG** Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

26080 **LC\_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

26082 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

**26085 *LC\_MESSAGES***

26086 Determine the locale that should be used to affect the format and contents of  
26087 diagnostic messages written to standard error.

**26088 *LC\_NUMERIC***

26089 Determine the locale for selecting the radix character used when writing floating-  
26090 point formatted output.

26091 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**26092 ASYNCHRONOUS EVENTS**

26093 Default.

**26094 STDOUT**

26095 See the EXTENDED DESCRIPTION section.

**26096 STDERR**

26097 The standard error shall be used only for diagnostic messages.

**26098 OUTPUT FILES**

26099 None.

**26100 EXTENDED DESCRIPTION**

26101 The *od* utility shall copy sequentially each input file to standard output, transforming the input  
26102 XSI data according to the output types specified by the **-t** option or the **-bcdosx** options. If no  
26103 output type is specified, the default output shall be as if **-t os** had been specified.

26104 The number of bytes transformed by the output type specifier **c** may be variable depending on  
26105 the *LC\_CTYPE* category.

26106 The default number of bytes transformed by output type specifiers **d**, **f**, **o**, **u**, and **x** corresponds  
26107 to the various C-language types as follows. If the **c99** compiler is present on the system, these  
26108 specifiers shall correspond to the sizes used by default in that compiler. Otherwise, these sizes  
26109 may vary among systems that conform to IEEE Std 1003.1-2001.

- 26110 • For the type specifier characters **d**, **o**, **u**, and **x**, the default number of bytes shall correspond  
26111 to the size of the underlying implementation's basic integer type. For these specifier  
26112 characters, the implementation shall support values of the optional number of bytes to be  
26113 converted corresponding to the number of bytes in the C-language types **char**, **short**, **int**, and  
26114 **long**. These numbers can also be specified by an application as the characters '**C**', '**S**', '**I**',  
26115 and '**L**', respectively. The implementation shall also support the values 1, 2, 4, and 8, even if  
26116 it provides no C-Language types of those sizes. The implementation shall support the

26117 decimal value corresponding to the C-language type **long long**. The byte order used when  
 26118 interpreting numeric values is implementation-defined, but shall correspond to the order in  
 26119 which a constant of the corresponding type is stored in memory on the system.

- 26120 • For the type specifier character **f**, the default number of bytes shall correspond to the number  
 26121 of bytes in the underlying implementation's basic double precision floating-point data type.  
 26122 The implementation shall support values of the optional number of bytes to be converted  
 26123 corresponding to the number of bytes in the C-language types **float**, **double**, and **long**  
 26124 **double**. These numbers can also be specified by an application as the characters '**F**', '**D**',  
 26125 and '**L**', respectively.

26126 The type specifier character **a** specifies that bytes shall be interpreted as named characters from  
 26127 the International Reference Version (IRV) of the ISO/IEC 646:1991 standard. Only the least  
 26128 significant seven bits of each byte shall be used for this type specification. Bytes with the values  
 26129 listed in the following table shall be written using the corresponding names for those characters.

26130 **Table 4-12** Named Characters in *od*

| Value | Name       | Value | Name       | Value | Name                         | Value | Name       |
|-------|------------|-------|------------|-------|------------------------------|-------|------------|
| \000  | <b>nul</b> | \001  | <b>soh</b> | \002  | <b>stx</b>                   | \003  | <b>etx</b> |
| \004  | <b>eot</b> | \005  | <b>enq</b> | \006  | <b>ack</b>                   | \007  | <b>bel</b> |
| \010  | <b>bs</b>  | \011  | <b>ht</b>  | \012  | <b>If or nl</b> <sup>*</sup> | \013  | <b>vt</b>  |
| \014  | <b>ff</b>  | \015  | <b>cr</b>  | \016  | <b>so</b>                    | \017  | <b>si</b>  |
| \020  | <b>dle</b> | \021  | <b>dcl</b> | \022  | <b>dc2</b>                   | \023  | <b>de3</b> |
| \024  | <b>dc4</b> | \025  | <b>nak</b> | \026  | <b>syn</b>                   | \027  | <b>etb</b> |
| \030  | <b>can</b> | \031  | <b>em</b>  | \032  | <b>sub</b>                   | \033  | <b>esc</b> |
| \034  | <b>fs</b>  | \035  | <b>gs</b>  | \036  | <b>rs</b>                    | \037  | <b>us</b>  |
| \040  | <b>sp</b>  | \177  | <b>del</b> |       |                              |       |            |

26142 **Note:** The "\012" value may be written either as **If** or **nl**.

26143 The type specifier character **c** specifies that bytes shall be interpreted as characters specified by  
 26144 the current setting of the *LC\_CTYPE* locale category. Characters listed in the table in the Base  
 26145 Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation ('\\', '\a', '\b',  
 26146 '\f', '\n', '\r', '\t', '\v') shall be written as the corresponding escape sequences, except  
 26147 that backslash shall be written as a single backslash and a NUL shall be written as '\0'. Other  
 26148 non-printable characters shall be written as one three-digit octal number for each byte in the  
 26149 character. If the size of a byte on the system is greater than nine bits, the format used for non-  
 26150 printable characters is implementation-defined. Printable multi-byte characters shall be written  
 26151 in the area corresponding to the first byte of the character; the two-character sequence " \*\*"  
 26152 shall be written in the area corresponding to each remaining byte in the character, as an  
 26153 indication that the character is continued. When either the **-j skip** or **-N count** option is specified  
 26154 along with the **c** type specifier, and this results in an attempt to start or finish in the middle of a  
 26155 multi-byte character, the result is implementation-defined.

26156 The input data shall be manipulated in blocks, where a block is defined as a multiple of the least  
 26157 common multiple of the number of bytes transformed by the specified output types. If the least  
 26158 common multiple is greater than 16, the results are unspecified. Each input block shall be  
 26159 written as transformed by each output type, one per written line, in the order that the output  
 26160 types were specified. If the input block size is larger than the number of bytes transformed by  
 26161 the output type, the output type shall sequentially transform the parts of the input block, and  
 26162 the output from each of the transformations shall be separated by one or more <blank>s.

26163 If, as a result of the specification of the **-N** option or end-of-file being reached on the last input  
26164 file, input data only partially satisfies an output type, the input shall be extended sufficiently  
26165 with null bytes to write the last byte of the input.

26166 Unless **-A n** is specified, the first output line produced for each input block shall be preceded by  
26167 the input offset, cumulative across input files, of the next byte to be written. The format of the  
26168 input offset is unspecified; however, it shall not contain any <blank>s, shall start at the first  
26169 character of the output line, and shall be followed by one or more <blank>s. In addition, the  
26170 offset of the byte following the last byte written shall be written after all the input data has been  
26171 processed, but shall not be followed by any <blank>s.

26172 If no **-A** option is specified, the input offset base is unspecified.

### 26173 EXIT STATUS

26174 The following exit values shall be returned:

26175 0 All input files were processed successfully.  
26176 >0 An error occurred.

### 26177 CONSEQUENCES OF ERRORS

26178 Default.

### 26179 APPLICATION USAGE

26180 XSI-conformant applications are warned not to use filenames starting with '+' or a first  
26181 operand starting with a numeric character so that the old functionality can be maintained by  
26182 implementations, unless they specify one of the **-A**, **-J**, or **-N** options. To guarantee that one of  
26183 these filenames is always interpreted as a filename, an application could always specify the  
26184 address base format with the **-A** option.

### 26185 EXAMPLES

26186 If a file containing 128 bytes with decimal values zero to 127, in increasing order, is supplied as  
26187 standard input to the command:

26188 `od -A d -t a`

26189 on an implementation using an input block size of 16 bytes, the standard output, independent of  
26190 the current locale setting, would be similar to:

```
26191 0000000 nul soh stx etx eot enq ack bel bs ht nl vt ff cr so si
26192 0000016 dle dc1 dc2 dc3 dc4 nak syn etb can em sub esc fs gs rs us
26193 0000032 sp ! " # $ % & ' () * + , - . /
26194 0000048 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
26195 0000064 @ A B C D E F G H I J K L M N O
26196 0000080 P Q R S T U V W X Y Z [\] ^ _
26197 0000096 ` a b c d e f g h i j k l m n o
26198 0000112 p q r s t u v w x y z { | } ~ del
26199 0000128
```

26200 Note that this volume of IEEE Std 1003.1-2001 allows **nl** or **If** to be used as the name for the  
26201 ISO/IEC 646: 1991 standard IRV character with decimal value 10. The IRV names this character  
26202 **If** (line feed), but traditional implementations have referred to this character as newline (**nl**) and  
26203 the POSIX locale character set symbolic name for the corresponding character is a <newline>.

26204 The command:

26205 `od -A o -t o2x2x -n 18`

26206 on a system with 32-bit words and an implementation using an input block size of 16 bytes  
26207 could write 18 bytes in approximately the following format:

```
26208 0000000 032056 031440 041123 042040 052516 044530 020043 031464
26209 342e 3320 4253 4420 554e 4958 2023 3334
26210 342e3320 42534420 554e4958 20233334
26211 0000020 032472
26212 353a
26213 353a0000
26214 0000022
26215 The command:
26216 od -A d -t f -t o4 -t x4 -n 24 -j 0x15
26217 on a system with 64-bit doubles (for example, IEEE Std 754-1985 double precision floating-point
26218 format) would skip 21 bytes of input data and then write 24 bytes in approximately the
26219 following format:
26220 0000000 1.00000000000000e+00 1.57350000000000e+01
26221 07774000000 000000000000 10013674121 35341217270
26222 3ff00000 00000000 402f3851 eb851eb8
26223 0000016 1.40668230000000e+02
26224 10030312542 04370303230
26225 40619562 23e18698
26226 0000024
26227 RATIONALE
26228 The od utility went through several names in early proposals, including hd, xd, and most recently
26229 hexdump. There were several objections to all of these based on the following reasons:
26230 • The hd and xd names conflicted with historical utilities that behaved differently.
26231 • The hexdump description was much more complex than needed for a simple dump utility.
26232 • The od utility has been available on all historical implementations and there was no need to
26233 create a new name for a utility so similar to the historical od utility.
26234 The original reasons for not standardizing historical od were also fairly widespread. Those
26235 reasons are given below along with rationale explaining why the standard developers believe
26236 that this version does not suffer from the indicated problem:
26237 • The BSD and System V versions of od have diverged, and the intersection of features
26238 provided by both does not meet the needs of the user community. In fact, the System V
26239 version only provides a mechanism for dumping octal bytes and shorts, signed and unsigned
26240 decimal shorts, hexadecimal shorts, and ASCII characters. BSD added the ability to dump
26241 floats, doubles, named ASCII characters, and octal, signed decimal, unsigned decimal, and
26242 hexadecimal longs. The version presented here provides more normalized forms for
26243 dumping bytes, shorts, ints, and longs in octal, signed decimal, unsigned decimal, and
26244 hexadecimal; float, double, and long double; and named ASCII as well as current locale
26245 characters.
26246 • It would not be possible to come up with a compatible superset of the BSD and System V
26247 flags that met the requirements of the standard developers. The historical default od output is
26248 the specified default output of this utility. None of the option letters chosen for this version
26249 of od conflict with any of the options to historical versions of od.
26250 • On systems with different sizes for short, int, and long, there was no way to ask for dumps
26251 of ints, even in the BSD version. Because of the way options are named, the name space
26252 could not be extended to solve these problems. This is why the -t option was added (with
26253 type specifiers more closely matched to the printf() formats used in the rest of this volume of
```

IEEE Std 1003.1-2001) and the optional field sizes were added to the d, f, o, u, and x type specifiers. It is also one of the reasons why the historical practice was not mandated as a required obsolescent form of *od*. (Although the old versions of *od* are not listed as an obsolescent form, implementations are urged to continue to recognize the older forms for several more years.) The a, c, f, o, and x types match the meaning of the corresponding format characters in the historical implementations of *od* except for the default sizes of the fields converted. The d format is signed in this volume of IEEE Std 1003.1-2001 to match the *printf()* notation. (Historical versions of *od* used d as a synonym for u in this version. The System V implementation uses s for signed decimal; BSD uses i for signed decimal and s for null-terminated strings.) Other than d and u, all of the type specifiers match format characters in the historical BSD version of **od**.

The sizes of the C-language types **char**, **short**, **int**, **long**, **float**, **double**, and **long double** are used even though it is recognized that there may be zero or more than one compiler for the C language on an implementation and that they may use different sizes for some of these types. (For example, one compiler might use 2 bytes **shorts**, 2 bytes **ints**, and 4 bytes **longs**, while another compiler (or an option to the same compiler) uses 2 bytes **shorts**, 4 bytes **ints**, and 4 bytes **longs**.) Nonetheless, there has to be a basic size known by the implementation for these types, corresponding to the values reported by invocations of the *getconf* utility when called with *system\_var* operands {UCHAR\_MAX}, {USHORT\_MAX}, {UINT\_MAX}, and {ULONG\_MAX} for the types **char**, **short**, **int**, and **long**, respectively. There are similar constants required by the ISO C standard, but not required by the System Interfaces volume of IEEE Std 1003.1-2001 or this volume of IEEE Std 1003.1-2001. They are {FLT\_MANT\_DIG}, {DBL\_MANT\_DIG}, and {LDBL\_MANT\_DIG} for the types **float**, **double**, and **long double**, respectively. If the optional *c99* utility is provided by the implementation and used as specified by this volume of IEEE Std 1003.1-2001, these are the sizes that would be provided. If an option is used that specifies different sizes for these types, there is no guarantee that the *od* utility is able to interpret binary data output by such a program correctly.

This volume of IEEE Std 1003.1-2001 requires that the numeric values of these lengths be recognized by the *od* utility and that symbolic forms also be recognized. Thus, a conforming application can always look at an array of **unsigned long** data elements using *od -t uL*.

- The method of specifying the format for the address field based on specifying a starting offset in a file unnecessarily tied the two together. The **-A** option now specifies the address base and the **-S** option specifies a starting offset.
- It would be difficult to break the dependence on U.S. ASCII to achieve an internationalized utility. It does not seem to be any harder for *od* to dump characters in the current locale than it is for the *ed* or *sed l* commands. The c type specifier does this without difficulty and is completely compatible with the historical implementations of the c format character when the current locale uses a superset of the ISO/IEC 646:1991 standard as a codeset. The a type specifier (from the BSD a format character) was left as a portable means to dump ASCII (or more correctly ISO/IEC 646:1991 standard (IRV)) so that headers produced by *pax* could be deciphered even on systems that do not use the ISO/IEC 646:1991 standard as a subset of their base codeset.

The use of " \*\* " as an indication of continuation of a multi-byte character in c specifier output was chosen based on seeing an implementation that uses this method. The continuation bytes have to be marked in a way that is not ambiguous with another single-byte or multi-byte character.

An early proposal used **-S** and **-n**, respectively, for the **-j** and **-N** options eventually selected. These were changed to avoid conflicts with historical implementations.

26302        The original standard specified **-t o2** as the default when no output type was given. This was  
26303        changed to **-t oS** (the length of a **short**) to accommodate a supercomputer implementation that  
26304        historically used 64 bits as its default (and that defined shorts as 64 bits). This change should not  
26305        affect conforming applications. The requirement to support lengths of 1, 2, and 4 was added at  
26306        the same time to address an historical implementation that had no two-byte data types in its C  
26307        compiler.

26308        The use of a basic integer data type is intended to allow the implementation to choose a word  
26309        size commonly used by applications on that architecture.

#### 26310 FUTURE DIRECTIONS

26311        All option and operand interfaces marked as extensions may be withdrawn in a future version.

#### 26312 SEE ALSO

26313        *c99, sed*

#### 26314 CHANGE HISTORY

26315        First released in Issue 2.

#### 26316 Issue 5

26317        In the description of the **-c** option, the phrase “This is equivalent to **-t c.**” is deleted.

26318        The FUTURE DIRECTIONS section is modified.

#### 26319 Issue 6

26320        The *od* utility is changed to remove the assumption that **short** was a two-byte entity, as per the  
26321        revisions in the IEEE P1003.2b draft standard.

26322        The normative text is reworded to avoid use of the term “must” for application requirements.

## 26323 NAME

26324        paste — merge corresponding or subsequent lines of files

## 26325 SYNOPSIS

26326        paste [-s][-d *list*] *file*...

## 26327 DESCRIPTION

26328        The *paste* utility shall concatenate the corresponding lines of the given input files, and write the  
26329        resulting lines to standard output.26330        The default operation of *paste* shall concatenate the corresponding lines of the input files. The  
26331        <newline> of every line except the line from the last input file shall be replaced with a <tab>.26332        If an end-of-file condition is detected on one or more input files, but not all input files, *paste* shall  
26333        behave as though empty lines were read from the files on which end-of-file was detected, unless  
26334        the **-s** option is specified.

## 26335 OPTIONS

26336        The *paste* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
26337        12.2, Utility Syntax Guidelines.

26338        The following options shall be supported:

26339        **-d** *list*        Unless a backslash character appears in *list*, each character in *list* is an element  
26340        specifying a delimiter character. If a backslash character appears in *list*, the  
26341        backslash character and one or more characters following it are an element  
26342        specifying a delimiter character as described below. These elements specify one or  
26343        more delimiters to use, instead of the default <tab>, to replace the <newline> of  
26344        the input lines. The elements in *list* shall be used circularly; that is, when the list is  
26345        exhausted the first element from the list is reused. When the **-s** option is specified:

- 26346        • The last <newline> in a file shall not be modified.
- 26347        • The delimiter shall be reset to the first element of *list* after each *file* operand is  
26348        processed.

26349        When the **-s** option is not specified:

- 26350        • The <newline>s in the file specified by the last *file* operand shall not be  
26351        modified.
- 26352        • The delimiter shall be reset to the first element of *list* each time a line is  
26353        processed from each file.

26354        If a backslash character appears in *list*, it and the character following it shall be  
26355        used to represent the following delimiter characters:

26356        \n &lt;newline&gt;.

26357        \t &lt;tab&gt;.

26358        \\ Backslash character.

26359        \0 Empty string (not a null character). If '\0' is immediately followed by the  
26360        character 'x', the character 'X', or any character defined by the *LC\_CTYPE*  
26361        **digit** keyword (see the Base Definitions volume of IEEE Std 1003.1-2001,  
26362        Chapter 7, Locale), the results are unspecified.

26363        If any other characters follow the backslash, the results are unspecified.

26364        **-s**        Concatenate all of the lines of each separate input file in command line order. The  
26365        <newline> of every line except the last line in each input file shall be replaced with

26366 the <tab>, unless otherwise specified by the **-d** option.

## 26367 OPERANDS

26368 The following operand shall be supported:

26369 *file* A pathname of an input file. If ‘–’ is specified for one or more of the *files*, the  
26370 standard input shall be used; the standard input shall be read one line at a time,  
26371 circularly, for each instance of ‘–’. Implementations shall support pasting of at  
26372 least 12 *file* operands.

## 26373 STDIN

26374 The standard input shall be used only if one or more *file* operands is ‘–’. See the INPUT FILES  
26375 section.

## 26376 INPUT FILES

26377 The input files shall be text files, except that line lengths shall be unlimited.

## 26378 ENVIRONMENT VARIABLES

26379 The following environment variables shall affect the execution of *paste*:

26380 *LANG* Provide a default value for the internationalization variables that are unset or null.  
26381 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
26382 Internationalization Variables for the precedence of internationalization variables  
26383 used to determine the values of locale categories.)

26384 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
26385 internationalization variables.

26386 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
26387 characters (for example, single-byte as opposed to multi-byte characters in  
26388 arguments and input files).

### 26389 *LC\_MESSAGES*

26390 Determine the locale that should be used to affect the format and contents of  
26391 diagnostic messages written to standard error.

26392 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

## 26393 ASYNCHRONOUS EVENTS

26394 Default.

## 26395 STDOUT

26396 Concatenated lines of input files shall be separated by the <tab> (or other characters under the  
26397 control of the **-d** option) and terminated by a <newline>.

## 26398 STDERR

26399 The standard error shall be used only for diagnostic messages.

## 26400 OUTPUT FILES

26401 None.

## 26402 EXTENDED DESCRIPTION

26403 None.

## 26404 EXIT STATUS

26405 The following exit values shall be returned:

26406 0 Successful completion.

26407 >0 An error occurred.

## 26408 CONSEQUENCES OF ERRORS

26409 If one or more input files cannot be opened when the **-s** option is not specified, a diagnostic  
26410 message shall be written to standard error, but no output is written to standard output. If the **-s**  
26411 option is specified, the *paste* utility shall provide the default behavior described in Section 1.11  
26412 (on page 20).

## 26413 APPLICATION USAGE

26414 When the escape sequences of the *list* option-argument are used in a shell script, they must be  
26415 quoted; otherwise, the shell treats the '\' as a special character.

26416 Conforming applications should only use the specific backslash escaped delimiters presented in  
26417 this volume of IEEE Std 1003.1-2001. Historical implementations treat '\x', where 'x' is not in  
26418 this list, as 'x', but future implementations are free to expand this list to recognize other  
26419 common escapes similar to those accepted by *printf* and other standard utilities.

26420 Most of the standard utilities work on text files. The *cut* utility can be used to turn files with  
26421 arbitrary line lengths into a set of text files containing the same data. The *paste* utility can be used  
26422 to create (or recreate) files with arbitrary line lengths. For example, if *file* contains long lines:

```
26423 cut -b 1-500 -n file > file1
26424 cut -b 501- -n file > file2
```

26425 creates **file1** (a text file) with lines no longer than 500 bytes (plus the <newline>) and **file2** that  
26426 contains the remainder of the data from *file*. Note that **file2** is not a text file if there are lines in  
26427 *file* that are longer than 500 + {LINE\_MAX} bytes. The original file can be recreated from **file1**  
26428 and **file2** using the command:

```
26429 paste -d "\0" file1 file2 > file
```

26430 The commands:

```
26431 paste -d "\0" ...
26432 paste -d "" ...
```

26433 are not necessarily equivalent; the latter is not specified by this volume of IEEE Std 1003.1-2001  
26434 and may result in an error. The construct '\0' is used to mean "no separator" because  
26435 historical versions of *paste* did not follow the syntax guidelines, and the command:

```
26436 paste -d "" ...
```

26437 could not be handled properly by *getopt()*.

## 26438 EXAMPLES

26439 1. Write out a directory in four columns:

```
26440 ls | paste - - -
```

26441 2. Combine pairs of lines from a file into single lines:

```
26442 paste -s -d "\t\n" file
```

## 26443 RATIONALE

26444 None.

## 26445 FUTURE DIRECTIONS

26446 None.

**26447 SEE ALSO**

26448       Section 1.11 (on page 20), *cut*, *grep*, *pr*

**26449 CHANGE HISTORY**

26450       First released in Issue 2.

**26451 Issue 6**

26452       The normative text is reworded to avoid use of the term “must” for application requirements.

## 26453 NAME

26454 patch — apply changes to files

## 26455 SYNOPSIS

26456 UP patch [-blNR][ -c | -e | -n][-d dir][-D define][-i patchfile]  
26457 [-o outfile][-p num][-r rejectfile][file]

26458

## 26459 DESCRIPTION

26460 The *patch* utility shall read a source (patch) file containing any of the three forms of difference  
26461 (diff) listings produced by the *diff* utility (normal, context, or in the style of *ed*) and apply those  
26462 differences to a file. By default, *patch* shall read from the standard input.26463 The *patch* utility shall attempt to determine the type of the *diff* listing, unless overruled by a **-c**,  
26464 **-e**, or **-n** option.26465 If the patch file contains more than one patch, *patch* shall attempt to apply each of them as if they  
26466 came from separate patch files. (In this case, the application shall ensure that the name of the  
26467 patch file is determinable for each *diff* listing.)

## 26468 OPTIONS

26469 The *patch* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
26470 12.2, Utility Syntax Guidelines.

26471 The following options shall be supported:

26472 **-b** Save a copy of the original contents of each modified file, before the differences are  
26473 applied, in a file of the same name with the suffix **.orig** appended to it. If the file  
26474 already exists, it shall be overwritten; if multiple patches are applied to the same  
26475 file, the **.orig** file shall be written only for the first patch. When the **-o** *outfile* option  
26476 is also specified, *file.orig* shall not be created but, if *outfile* already exists,  
26477 *outfile.orig* shall be created.26478 **-c** Interpret the patch file as a context difference (the output of the utility *diff* when  
26479 the **-c** or **-C** options are specified).26480 **-d** *dir* Change the current directory to *dir* before processing as described in the  
26481 EXTENDED DESCRIPTION section.26482 **-D** *define* Mark changes with one of the following C preprocessor constructs:26483       **#ifdef** *define*

26484       ...

26485       **#endif**26486       **#ifndef** *define*

26487       ...

26488       **#endif**26489       optionally combined with the C preprocessor construct **#else**.26490 **-e** Interpret the patch file as an *ed* script, rather than a *diff* script.26491 **-i** *patchfile* Read the patch information from the file named by the pathname *patchfile*, rather  
26492 than the standard input.26493 **-l** (The letter ell.) Cause any sequence of <blank>s in the difference script to match  
26494 any sequence of <blank>s in the input file. Other characters shall be matched  
26495 exactly.

|       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26496 | <b>-n</b>            | Interpret the script as a normal difference.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26497 | <b>-N</b>            | Ignore patches where the differences have already been applied to the file; by default, already-applied patches shall be rejected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26499 | <b>-o outfile</b>    | Instead of modifying the files (specified by the <i>file</i> operand or the difference listings) directly, write a copy of the file referenced by each patch, with the appropriate differences applied, to <i>outfile</i> . Multiple patches for a single file shall be applied to the intermediate versions of the file created by any previous patches, and shall result in multiple, concatenated versions of the file being written to <i>outfile</i> .                                                                                                                                                                                                                    |
| 26505 | <b>-p num</b>        | For all pathnames in the patch file that indicate the names of files to be patched, delete <i>num</i> pathname components from the beginning of each pathname. If the pathname in the patch file is absolute, any leading slashes shall be considered the first component (that is, <b>-p 1</b> shall remove the leading slashes). Specifying <b>-p 0</b> shall cause the full pathname to be used. If <b>-p</b> is not specified, only the basename (the final pathname component) shall be used.                                                                                                                                                                             |
| 26511 | <b>-R</b>            | Reverse the sense of the patch script; that is, assume that the difference script was created from the new version to the old version. The <b>-R</b> option cannot be used with <i>ed</i> scripts. The <i>patch</i> utility shall attempt to reverse each portion of the script before applying it. Rejected differences shall be saved in swapped format. If this option is not specified, and until a portion of the patch file is successfully applied, <i>patch</i> attempts to apply each portion in its reversed sense as well as in its normal sense. If the attempt is successful, the user shall be prompted to determine whether the <b>-R</b> option should be set. |
| 26519 | <b>-r rejectfile</b> | Override the default reject filename. In the default case, the reject file shall have the same name as the output file, with the suffix <b>.rej</b> appended to it; see <b>Patch Application</b> (on page 691).                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 26522 OPERANDS

26523 The following operand shall be supported:

26524 *file* A pathname of a file to patch.

## 26525 STDIN

26526 See the INPUT FILES section.

## 26527 INPUT FILES

26528 Input files shall be text files.

## 26529 ENVIRONMENT VARIABLES

26530 The following environment variables shall affect the execution of *patch*:

|       |                 |                                                                                                                                                                                                                                                                                                       |
|-------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26531 | <b>LANG</b>     | Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.) |
| 26535 | <b>LC_ALL</b>   | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                              |
| 26537 | <b>LC_CTYPE</b> | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).                                                                                                             |

|           |                               |                                                                                                                                                                                                                                                                                                                                      |
|-----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26540     | <b>LC_MESSAGES</b>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.                                                                                                                                                     |
| 26541     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26542     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26543     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26544 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                |
| 26545     | <b>LC_TIME</b>                | Determine the locale for recognizing the format of file timestamps written by the <i>diff</i> utility in a context-difference input file.                                                                                                                                                                                            |
| 26546     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26547     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                                                                                      |
| 26548     |                               | Default.                                                                                                                                                                                                                                                                                                                             |
| 26549     | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                                                                                      |
| 26550     |                               | Not used.                                                                                                                                                                                                                                                                                                                            |
| 26551     | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                                                                                      |
| 26552     |                               | The standard error shall be used for diagnostic and informational messages.                                                                                                                                                                                                                                                          |
| 26553     | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                      |
| 26554     |                               | The output of the <i>patch</i> utility, the save files ( <b>.orig</b> suffixes), and the reject files ( <b>.rej</b> suffixes) shall be text files.                                                                                                                                                                                   |
| 26555     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26556     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                                                                                      |
| 26557     |                               | A patch file may contain patching instructions for more than one file; filenames shall be determined as specified in <b>Filename Determination</b> (on page 691). When the <b>-b</b> option is specified, for each patched file, the original shall be saved in a file of the same name with the suffix <b>.orig</b> appended to it. |
| 26558     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26559     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26560     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26561     |                               | For each patched file, a reject file may also be created as noted in <b>Patch Application</b> (on page 691). In the absence of a <b>-r</b> option, the name of this file shall be formed by appending the suffix <b>.rej</b> to the original filename.                                                                               |
| 26562     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26563     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26564     | <b>Patch File Format</b>      |                                                                                                                                                                                                                                                                                                                                      |
| 26565     |                               | The patch file shall contain zero or more lines of header information followed by one or more patches. Each patch shall contain zero or more lines of filename identification in the format produced by <i>diff -c</i> , and one or more sets of <i>diff</i> output, which are customarily called <i>hunks</i> .                     |
| 26566     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26567     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26568     |                               | The <i>patch</i> utility shall recognize the following expression in the header information:                                                                                                                                                                                                                                         |
| 26569     | <b>Index: pathname</b>        |                                                                                                                                                                                                                                                                                                                                      |
| 26570     |                               | The file to be patched is named <i>pathname</i> .                                                                                                                                                                                                                                                                                    |
| 26571     |                               | If all lines (including headers) within a patch begin with the same leading sequence of <blank>s, the <i>patch</i> utility shall remove this sequence before proceeding. Within each patch, if the type of difference is context, the <i>patch</i> utility shall recognize the following expressions:                                |
| 26572     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26573     |                               |                                                                                                                                                                                                                                                                                                                                      |
| 26574     | <b>*** filename timestamp</b> |                                                                                                                                                                                                                                                                                                                                      |
| 26575     |                               | The patches arose from <i>filename</i> .                                                                                                                                                                                                                                                                                             |
| 26576     | <b>-- filename timestamp</b>  |                                                                                                                                                                                                                                                                                                                                      |
| 26577     |                               | The patches should be applied to <i>filename</i> .                                                                                                                                                                                                                                                                                   |
| 26578     |                               | Each hunk within a patch shall be the <i>diff</i> output to change a line range within the original file.                                                                                                                                                                                                                            |
| 26579     |                               | The line numbers for successive hunks within a patch shall occur in ascending order.                                                                                                                                                                                                                                                 |

26580 **Filename Determination**

26581 If no *file* operand is specified, *patch* shall perform the following steps to determine the filename  
26582 to use:

- 26583 1. If the type of *diff* is context, the *patch* utility shall delete pathname components (as  
26584 specified by the **-p** option) from the filename on the line beginning with " \*\*\* ", then test  
26585 for the existence of this file relative to the current directory (or the directory specified with  
26586 the **-d** option). If the file exists, the *patch* utility shall use this filename.
- 26587 2. If the type of *diff* is context, the *patch* utility shall delete the pathname components (as  
26588 specified by the **-p** option) from the filename on the line beginning with " --- ", then test  
26589 for the existence of this file relative to the current directory (or the directory specified with  
26590 the **-d** option). If the file exists, the *patch* utility shall use this filename.
- 26591 3. If the header information contains a line beginning with the string **Index:**, the *patch* utility  
26592 shall delete pathname components (as specified by the **-p** option) from this line, then test  
26593 for the existence of this file relative to the current directory (or the directory specified with  
26594 the **-d** option). If the file exists, the *patch* utility shall use this filename.
- 26595 XSI 4. If an **SCCS** directory exists in the current directory, *patch* shall attempt to perform a **get -e**  
26596 **SCCS/s.filename** command to retrieve an editable version of the file. If the file exists, the  
26597 *patch* utility shall use this filename.
- 26598 5. The *patch* utility shall write a prompt to standard output and request a filename  
26599 interactively from the controlling terminal (for example, **/dev/tty**).

26600 **Patch Application**

26601 If the **-c**, **-e**, or **-n** option is present, the *patch* utility shall interpret information within each hunk  
26602 as a context difference, an *ed* difference, or a normal difference, respectively. In the absence of  
26603 any of these options, the *patch* utility shall determine the type of difference based on the format  
26604 of information within the hunk.

26605 For each hunk, the *patch* utility shall begin to search for the place to apply the patch at the line  
26606 number at the beginning of the hunk, plus or minus any offset used in applying the previous  
26607 hunk. If lines matching the hunk context are not found, *patch* shall scan both forwards and  
26608 backwards at least 1 000 bytes for a set of lines that match the hunk context.

26609 If no such place is found and it is a context difference, then another scan shall take place,  
26610 ignoring the first and last line of context. If that fails, the first two and last two lines of context  
26611 shall be ignored and another scan shall be made. Implementations may search more extensively  
26612 for installation locations.

26613 If no location can be found, the *patch* utility shall append the hunk to the reject file. The rejected  
26614 hunk shall be written in context-difference format regardless of the format of the patch file. If the  
26615 input was a normal or *ed*-style difference, the reject file may contain differences with zero lines  
26616 of context. The line numbers on the hunks in the reject file may be different from the line  
26617 numbers in the patch file since they shall reflect the approximate locations for the failed hunks in  
26618 the new file rather than the old one.

26619 If the type of patch is an *ed* diff, the implementation may accomplish the patching by invoking  
26620 the *ed* utility.

26621 **EXIT STATUS**

26622 The following exit values shall be returned:

26623 0 Successful completion.

26624 1 One or more lines were written to a reject file.

26625 >1 An error occurred.

## 26626 CONSEQUENCES OF ERRORS

26627 Patches that cannot be correctly placed in the file shall be written to a reject file.

## 26628 APPLICATION USAGE

26629 The **-R** option does not work with *ed* scripts because there is too little information to reconstruct  
26630 the reverse operation.

26631 The **-p** option makes it possible to customize a patch file to local user directory structures  
26632 without manually editing the patch file. For example, if the filename in the patch file was:

26633 /curds/whey/src/blurf1/blurf1.c

26634 Setting **-p 0** gives the entire pathname unmodified; **-p 1** gives:

26635 curds/whey/src/blurf1/blurf1.c

26636 without the leading slash, **-p 4** gives:

26637 blurf1/blurf1.c

26638 and not specifying **-p** at all gives:

26639 blurf1.c .

## 26640 EXAMPLES

26641 None.

## 26642 RATIONALE

26643 Some of the functionality in historical *patch* implementations was not specified. The following  
26644 documents those features present in historical implementations that have not been specified.

26645 A deleted piece of functionality was the '+' pseudo-option allowing an additional set of options  
26646 and a patch file operand to be given. This was seen as being insufficiently useful to standardize.

26647 In historical implementations, if the string "Prereq:" appeared in the header, the *patch* utility  
26648 would search for the corresponding version information (the string specified in the header,  
26649 delimited by <blank>s or the beginning or end of a line or the file) anywhere in the original file.  
26650 This was deleted as too simplistic and insufficiently trustworthy a mechanism to standardize.  
26651 For example, if:

26652 Prereq: 1.2

26653 were in the header, the presence of a delimited 1.2 anywhere in the file would satisfy the  
26654 prerequisite.

26655 The following options were dropped from historical implementations of *patch* as insufficiently  
26656 useful to standardize:

26657 **-b** The **-b** option historically provided a method for changing the name extension of  
26658 the backup file from the default **.orig**. This option has been modified and retained  
26659 in this volume of IEEE Std 1003.1-2001.

26660 **-F** The **-F** option specified the number of lines of a context diff to ignore when  
26661 searching for a place to install a patch.

26662 **-f** The **-f** option historically caused *patch* not to request additional information from  
26663 the user.

- 26664       **-r**       The **-r** option historically provided a method of overriding the extension of the  
26665        reject file from the default **.rej**.
- 26666       **-s**       The **-s** option historically caused *patch* to work silently unless an error occurred.
- 26667       **-x**       The **-x** option historically set internal debugging flags.
- 26668       In some file system implementations, the saving of a **.orig** file may produce unwanted results. In  
26669       the case of 12, 13, or 14-character filenames (on file systems supporting 14-character maximum  
26670       filenames), the **.orig** file overwrites the new file. The reject file may also exceed this filename  
26671       limit. It was suggested, due to some historical practice, that a tilde ('~') suffix be used instead  
26672       of **.orig** and some other character instead of the **.rej** suffix. This was rejected because it is not  
26673       obvious to the user which file is which. The suffixes **.orig** and **.rej** are clearer and more  
26674       understandable.
- 26675       The **-b** option has the opposite sense in some historical implementations—do not save the **.orig**  
26676       file. The default case here is not to save the files, making *patch* behave more consistently with the  
26677       other standard utilities.
- 26678       The **-w** option in early proposals was changed to **-l** to match historical practice.
- 26679       The **-N** option was included because without it, a non-interactive application cannot reject  
26680       previously applied patches. For example, if a user is piping the output of *diff* into the *patch*  
26681       utility, and the user only wants to patch a file to a newer version non-interactively, the **-N**  
26682       option is required.
- 26683       Changes to the **-l** option description were proposed to allow matching across <newline>s in  
26684       addition to just <blank>s. Since this is not historical practice, and since some ambiguities could  
26685       result, it is suggested that future developments in this area utilize another option letter, such as  
26686       **-L**.
- 26687 **FUTURE DIRECTIONS**
- 26688       None.
- 26689 **SEE ALSO**
- 26690       *ed*, *diff*
- 26691 **CHANGE HISTORY**
- 26692       First released in Issue 4.
- 26693 **Issue 5**
- 26694       The FUTURE DIRECTIONS section is added.
- 26695 **Issue 6**
- 26696       This utility is marked as part of the User Portability Utilities option.
- 26697       The description of the **-D** option and the steps in **Filename Determination** (on page 691) are  
26698       changed to match historical practice as defined in the IEEE P1003.2b draft standard.
- 26699       The normative text is reworded to avoid use of the term “must” for application requirements.

## 26700 NAME

26701 pathchk — check pathnames

## 26702 SYNOPSIS

26703 pathchk [-p] pathname...

## 26704 DESCRIPTION

26705 The *pathchk* utility shall check that one or more pathnames are valid (that is, they could be used  
26706 to access or create a file without causing syntax errors) and portable (that is, no filename  
26707 truncation results). More extensive portability checks are provided by the -p option.

26708 By default, the *pathchk* utility shall check each component of each *pathname* operand based on the  
26709 underlying file system. A diagnostic shall be written for each *pathname* operand that:

- 26710 • Is longer than {PATH\_MAX} bytes (see **Pathname Variable Values** in the Base Definitions  
26711 volume of IEEE Std 1003.1-2001, Chapter 13, Headers, <limits.h>)
- 26712 • Contains any component longer than {NAME\_MAX} bytes in its containing directory
- 26713 • Contains any component in a directory that is not searchable
- 26714 • Contains any character in any component that is not valid in its containing directory

26715 The format of the diagnostic message is not specified, but shall indicate the error detected and  
26716 the corresponding *pathname* operand.

26717 It shall not be considered an error if one or more components of a *pathname* operand do not exist  
26718 as long as a file matching the pathname specified by the missing components could be created  
26719 that does not violate any of the checks specified above.

## 26720 OPTIONS

26721 The *pathchk* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
26722 12.2, Utility Syntax Guidelines.

26723 The following option shall be supported:

26724 -p Instead of performing checks based on the underlying file system, write a  
26725 diagnostic for each *pathname* operand that:

- 26726 • Is longer than {\_POSIX\_PATH\_MAX} bytes (see **Minimum Values** in the Base  
26727 Definitions volume of IEEE Std 1003.1-2001, Chapter 13, Headers, <limits.h>)
- 26728 • Contains any component longer than {\_POSIX\_NAME\_MAX} bytes
- 26729 • Contains any character in any component that is not in the portable filename  
26730 character set

## 26731 OPERANDS

26732 The following operand shall be supported:

26733 *pathname* A pathname to be checked.

## 26734 STDIN

26735 Not used.

## 26736 INPUT FILES

26737 None.

## 26738 ENVIRONMENT VARIABLES

26739 The following environment variables shall affect the execution of *pathchk*:

26740 LANG Provide a default value for the internationalization variables that are unset or null.  
26741 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,

|           |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26742     |                               | Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)                                                                                                                                                                                                                                                                                                                                    |
| 26744     | <i>LC_ALL</i>                 | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                                                                                                                   |
| 26746     | <i>LC_CTYPE</i>               | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                                                                                                                                  |
| 26749     | <i>LC_MESSAGES</i>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                               |
| 26752 XSI | <i>NLSPATH</i>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                      |
| 26753     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26754     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26755     | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26756     |                               | Not used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26757     | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26758     |                               | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26759     | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26760     |                               | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 26761     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26762     |                               | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 26763     | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26764     |                               | The following exit values shall be returned:                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 26765     | 0                             | All <i>pathname</i> operands passed all of the checks.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 26766     | >0                            | An error occurred.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 26767     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26768     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26769     | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26770     |                               | The <i>test</i> utility can be used to determine whether a given pathname names an existing file; it does not, however, give any indication of whether or not any component of the pathname was truncated in a directory where the <i>_POSIX_NO_TRUNC</i> feature is not in effect. The <i>pathchk</i> utility does not check for file existence; it performs checks to determine whether a pathname does exist or could be created with no pathname component truncation. |
| 26775     |                               | The <i>noclobber</i> option in the shell (see the <i>set</i> special built-in) can be used to atomically create a file. As with all file creation semantics in the System Interfaces volume of IEEE Std 1003.1-2001, it guarantees atomic creation, but still depends on applications to agree on conventions and cooperate on the use of files after they have been created.                                                                                              |
| 26779     | <b>EXAMPLES</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26780     |                               | To verify that all pathnames in an imported data interchange archive are legitimate and unambiguous on the current system:                                                                                                                                                                                                                                                                                                                                                 |
| 26782     |                               | pax -f archive   sed -e '/ == .* /s///'   xargs pathchk                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 26783     |                               | if [ \$? -eq 0 ]                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 26784     |                               | then                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 26785     |                               | pax -r -f archive                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

```

26786 else
26787 echo Investigate problems before importing files.
26788 exit 1
26789 fi

```

To verify that all files in the current directory hierarchy could be moved to any system conforming to the System Interfaces volume of IEEE Std 1003.1-2001 that also supports the *pax* utility:

```

26793 find . -print | xargs pathchk -p
26794 if [$? -eq 0]
26795 then
26796 pax -w -f archive .
26797 else
26798 echo Portable archive cannot be created.
26799 exit 1
26800 fi

```

To verify that a user-supplied pathname names a readable file and that the application can create a file extending the given path without truncation and without overwriting any existing file:

```

26803 case $- in
26804 *C*) reset="" ;;
26805 *) reset="set +C"
26806 set -C;;
26807 esac
26808 test -r "$path" && pathchk "$path.out" &&
26809 rm "$path.out" > "$path.out"
26810 if [$? -ne 0]; then
26811 printf "%s: %s not found or %s.out fails \
26812 creation checks.\n" $0 "$path" "$path"
26813 $reset # Reset the noclobber option in case a trap
26814 # on EXIT depends on it.
26815 exit 1
26816 fi
26817 $reset
26818 PROCESSING < "$path" > "$path.out"

```

The following assumptions are made in this example:

- 26820 1. **PROCESSING** represents the code that is used by the application to use **\$path** once it is  
26821 verified that **\$path.out** works as intended.
- 26822 2. The state of the *noclobber* option is unknown when this code is invoked and should be set  
26823 on exit to the state it was in when this code was invoked. (The **reset** variable is used in this  
26824 example to restore the initial state.)
- 26825 3. Note the usage of:

```
26826 rm "$path.out" > "$path.out"
```

- 26827 a. The *pathchk* command has already verified, at this point, that **\$path.out** is not  
26828 truncated.
- 26829 b. With the *noclobber* option set, the shell verifies that **\$path.out** does not already exist  
26830 before invoking *rm*.

- 26831       c. If the shell succeeded in creating `$path.out`, `rm` removes it so that the application can  
26832       create the file again in the **PROCESSING** step.  
26833       d. If the **PROCESSING** step wants the file to exist already when it is invoked, the:  
26834           `rm "$path.out" > "$path.out"`  
26835           should be replaced with:  
26836           `> "$path.out"`  
26837           which verifies that the file did not already exist, but leaves `$path.out` in place for use  
26838           by **PROCESSING**.

#### 26839 RATIONALE

26840       The `pathchk` utility was new for the ISO POSIX-2:1993 standard. It, along with the `set`  
26841       `-C(noclobber)` option added to the shell, replaces the `mktemp`, `validfnam`, and `create` utilities that  
26842       appeared in early proposals. All of these utilities were attempts to solve several common  
26843       problems:

- 26844       • Verify the validity (for several different definitions of “valid”) of a pathname supplied by a  
26845        user, generated by an application, or imported from an external source.
- 26846       • Atomically create a file.
- 26847       • Perform various string handling functions to generate a temporary filename.

26848       The `create` utility, included in an early proposal, provided checking and atomic creation in a  
26849       single invocation of the utility; these are orthogonal issues and need not be grouped into a single  
26850       utility. Note that the `noclobber` option also provides a way of creating a lock for process  
26851       synchronization; since it provides an atomic `create`, there is no race between a test for existence  
26852       and the following creation if it did not exist.

26853       Having a function like `tmpnam()` in the ISO C standard is important in many high-level  
26854       languages. The shell programming language, however, has built-in string manipulation  
26855       facilities, making it very easy to construct temporary filenames. The names needed obviously  
26856       depend on the application, but are frequently of a form similar to:

26857       `$TMPDIR/application_abbreviation$$.suffix`

26858       In cases where there is likely to be contention for a given suffix, a simple shell **for** or **while** loop  
26859       can be used with the shell `noclobber` option to create a file without risk of collisions, as long as  
26860       applications trying to use the same filename name space are cooperating on the use of files after  
26861       they have been created.

#### 26862 FUTURE DIRECTIONS

26863       None.

#### 26864 SEE ALSO

26865       Section 2.7 (on page 43), `set` (on page 85), `test`

#### 26866 CHANGE HISTORY

26867       First released in Issue 4.

## 26868 NAME

26869 pax — portable archive interchange

## 26870 SYNOPSIS

```
26871 pax [-cdnv][-H|-L][-f archive][-s replstr]...[pattern...]
26872 pax -r[-cdiknuv][-H|-L][-f archive][-o options]...[-p string]...
26873 [-s replstr]...[pattern...]
26874 pax -w[-dituvX][-H|-L][-b blocksize][[-a][-f archive][-o options]...
26875 [-s replstr]...[-x format][file...]
26876 pax -r -w[-diklntuvX][-H|-L][-p string]...[-s replstr]...
26877 [file...] directory
```

## 26878 DESCRIPTION

26879 The *pax* utility shall read, write, and write lists of the members of archive files and copy  
26880 directory hierarchies. A variety of archive formats shall be supported; see the **-x format** option.

26881 The action to be taken depends on the presence of the **-r** and **-w** options. The four combinations  
26882 of **-r** and **-w** are referred to as the four modes of operation: **list**, **read**, **write**, and **copy** modes,  
26883 corresponding respectively to the four forms shown in the SYNOPSIS section.

26884 **list** In **list** mode (when neither **-r** nor **-w** are specified), *pax* shall write the names of  
26885 the members of the archive file read from the standard input, with pathnames  
26886 matching the specified patterns, to standard output. If a named file is of type  
26887 directory, the file hierarchy rooted at that file shall be listed as well.

26888 **read** In **read** mode (when **-r** is specified, but **-w** is not), *pax* shall extract the members of  
26889 the archive file read from the standard input, with pathnames matching the  
26890 specified patterns. If an extracted file is of type directory, the file hierarchy rooted  
26891 at that file shall be extracted as well. The extracted files shall be created performing  
26892 pathname resolution with the directory in which *pax* was invoked as the current  
26893 working directory.

26894 If an attempt is made to extract a directory when the directory already exists, this  
26895 shall not be considered an error. If an attempt is made to extract a FIFO when the  
26896 FIFO already exists, this shall not be considered an error.

26897 The ownership, access, and modification times, and file mode of the restored files  
26898 are discussed under the **-p** option.

26899 **write** In **write** mode (when **-w** is specified, but **-r** is not), *pax* shall write the contents of  
26900 the *file* operands to the standard output in an archive format. If no *file* operands are  
26901 specified, a list of files to copy, one per line, shall be read from the standard input.  
26902 A file of type directory shall include all of the files in the file hierarchy rooted at the  
26903 file.

26904 **copy** In **copy** mode (when both **-r** and **-w** are specified), *pax* shall copy the *file* operands  
26905 to the destination directory.

26906 If no *file* operands are specified, a list of files to copy, one per line, shall be read  
26907 from the standard input. A file of type directory shall include all of the files in the  
26908 file hierarchy rooted at the file.

26909 The effect of the **copy** shall be as if the copied files were written to an archive file  
26910 and then subsequently extracted, except that there may be hard links between the  
26911 original and the copied files. If the destination directory is a subdirectory of one of  
26912 the files to be copied, the results are unspecified. If the destination directory is a

26913 file of a type not defined by the System Interfaces volume of IEEE Std 1003.1-2001,  
26914 the results are implementation-defined; otherwise, it shall be an error for the file  
26915 named by the *directory* operand not to exist, not be writable by the user, or not be a  
26916 file of type directory.

26917 In **read** or **copy** modes, if intermediate directories are necessary to extract an archive member,  
26918 *pax* shall perform actions equivalent to the *mkdir()* function defined in the System Interfaces  
26919 volume of IEEE Std 1003.1-2001, called with the following arguments:

- 26920 • The intermediate directory used as the *path* argument
- 26921 • The value of the bitwise-inclusive OR of S\_IRWXU, S\_IRWXG, and S\_IRWXO as the *mode*  
26922 argument

26923 If any specified *pattern* or *file* operands are not matched by at least one file or archive member,  
26924 *pax* shall write a diagnostic message to standard error for each one that did not match and exit  
26925 with a non-zero exit status.

26926 The archive formats described in the EXTENDED DESCRIPTION section shall be automatically  
26927 detected on input. The default output archive format shall be implementation-defined.

26928 A single archive can span multiple files. The *pax* utility shall determine, in an implementation-  
26929 defined manner, what file to read or write as the next file.

26930 If the selected archive format supports the specification of linked files, it shall be an error if these  
26931 files cannot be linked when the archive is extracted. For archive formats that do not store file  
26932 contents with each name that causes a hard link, if the file that contains the data is not extracted  
26933 during this *pax* session, either the data shall be restored from the original file, or a diagnostic  
26934 message shall be displayed with the name of a file that can be used to extract the data. In  
26935 traversing directories, *pax* shall detect infinite loops; that is, entering a previously visited  
26936 directory that is an ancestor of the last file visited. When it detects an infinite loop, *pax* shall  
26937 write a diagnostic message to standard error and shall terminate.

## 26938 OPTIONS

26939 The *pax* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
26940 12.2, Utility Syntax Guidelines, except that the order of presentation of the **-o**, **-p**, and **-s** options  
26941 is significant.

26942 The following options shall be supported:

- 26943 **-r** Read an archive file from standard input.
- 26944 **-w** Write files to the standard output in the specified archive format.
- 26945 **-a** Append files to the end of the archive. It is implementation-defined which devices  
26946 on the system support appending. Additional file formats unspecified by this  
26947 volume of IEEE Std 1003.1-2001 may impose restrictions on appending.
- 26948 **-b blocksize** Block the output at a positive decimal integer number of bytes per write to the  
26949 archive file. Devices and archive formats may impose restrictions on blocking.  
26950 Blocking shall be automatically determined on input. Conforming applications  
26951 shall not specify a *blocksize* value larger than 32 256. Default blocking when  
26952 creating archives depends on the archive format. (See the **-x** option below.)
- 26953 **-c** Match all file or archive members except those specified by the *pattern* or *file*  
26954 operands.
- 26955 **-d** Cause files of type directory being copied or archived or archive members of type  
26956 directory being extracted or listed to match only the file or archive member itself  
26957 and not the file hierarchy rooted at the file.

|       |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26958 | <b>-f archive</b> | Specify the pathname of the input or output archive, overriding the default standard input (in <b>list</b> or <b>read</b> modes) or standard output ( <b>write</b> mode).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 26960 | <b>-H</b>         | If a symbolic link referencing a file of type directory is specified on the command line, <i>pax</i> shall archive the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. Otherwise, if a symbolic link referencing a file of any other file type which <i>pax</i> can normally archive is specified on the command line, then <i>pax</i> shall archive the file referenced by the link, using the name of the link. The default behavior shall be to archive the symbolic link itself.                                                                                                                                                                                                                                                                           |
| 26967 | <b>-i</b>         | Interactively rename files or archive members. For each archive member matching a <i>pattern</i> operand or file matching a <i>file</i> operand, a prompt shall be written to the file <b>/dev/tty</b> . The prompt shall contain the name of the file or archive member, but the format is otherwise unspecified. A line shall then be read from <b>/dev/tty</b> . If this line is blank, the file or archive member shall be skipped. If this line consists of a single period, the file or archive member shall be processed with no modification to its name. Otherwise, its name shall be replaced with the contents of the line. The <i>pax</i> utility shall immediately exit with a non-zero exit status if end-of-file is encountered when reading a response or if <b>/dev/tty</b> cannot be opened for reading and writing. |
| 26977 |                   | The results of extracting a hard link to a file that has been renamed during extraction are unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 26978 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 26979 | <b>-k</b>         | Prevent the overwriting of existing files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26980 | <b>-l</b>         | (The letter ell.) In <b>copy</b> mode, hard links shall be made between the source and destination file hierarchies whenever possible. If specified in conjunction with <b>-H</b> or <b>-L</b> , when a symbolic link is encountered, the hard link created in the destination file hierarchy shall be to the file referenced by the symbolic link. If specified when neither <b>-H</b> nor <b>-L</b> is specified, when a symbolic link is encountered, the implementation shall create a hard link to the symbolic link in the source file hierarchy or copy the symbolic link to the destination.                                                                                                                                                                                                                                   |
| 26987 | <b>-L</b>         | If a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, <i>pax</i> shall archive the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. Otherwise, if a symbolic link referencing a file of any other file type which <i>pax</i> can normally archive is specified on the command line or encountered during the traversal of a file hierarchy, <i>pax</i> shall archive the file referenced by the link, using the name of the link. The default behavior shall be to archive the symbolic link itself.                                                                                                                                                                |
| 26995 | <b>-n</b>         | Select the first archive member that matches each <i>pattern</i> operand. No more than one archive member shall be matched for each pattern (although members of type directory shall still match the file hierarchy rooted at that file).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26996 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 26997 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 26998 | <b>-o options</b> | Provide information to the implementation to modify the algorithm for extracting or writing files. The value of <i>options</i> shall consist of one or more comma-separated keywords of the form:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27001 |                   | <i>keyword</i> [[[:]= <i>value</i> ] [, <i>keyword</i> [[[:]= <i>value</i> ] , . . . ]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27002 |                   | Some keywords apply only to certain file formats, as indicated with each description. Use of keywords that are inapplicable to the file format being processed produces undefined results.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 27003 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 27004 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

27005           Keywords in the *options* argument shall be a string that would be a valid portable  
 27006           filename as described in the Base Definitions volume of IEEE Std 1003.1-2001,  
 27007           Section 3.276, Portable Filename Character Set.

27008           **Note:**       Keywords are not expected to be filenames, merely to follow the same character  
 27009           composition rules as portable filenames.

27010           Keywords can be preceded with white space. The *value* field shall consist of zero or  
 27011           more characters; within *value*, the application shall precede any literal comma with  
 27012           a backslash, which shall be ignored, but preserves the comma as part of *value*. A  
 27013           comma as the final character, or a comma followed solely by white space as the  
 27014           final characters, in *options* shall be ignored. Multiple **-o** options can be specified; if  
 27015           keywords given to these multiple **-o** options conflict, the keywords and values  
 27016           appearing later in command line sequence shall take precedence and the earlier  
 27017           shall be silently ignored. The following keyword values of *options* shall be  
 27018           supported for the file formats as indicated:

27019           **delete=pattern**

27020           (Applicable only to the **-x pax** format.) When used in **write** or **copy** mode, *pax*  
 27021           shall omit from extended header records that it produces any keywords  
 27022           matching the string pattern. When used in **read** or **list** mode, *pax* shall ignore  
 27023           any keywords matching the string pattern in the extended header records. In  
 27024           both cases, matching shall be performed using the pattern matching notation  
 27025           described in Section 2.13.1 (on page 62) and Section 2.13.2 (on page 63). For  
 27026           example:

27027           **-o delete=security.\***

27028           would suppress security-related information. See **pax Extended Header** (on  
 27029           page 711) for extended header record keyword usage.

27030           **exthdr.name=string**

27031           (Applicable only to the **-x pax** format.) This keyword allows user control over  
 27032           the name that is written into the **ustar** header blocks for the extended header  
 27033           produced under the circumstances described in **pax Header Block** (on page  
 27034           710). The name shall be the contents of *string*, after the following character  
 27035           substitutions have been made:

| <i>string</i><br>Includes: | Replaced By:                                                                                                       |
|----------------------------|--------------------------------------------------------------------------------------------------------------------|
| %d                         | The directory name of the file, equivalent to the result of the <i>dirname</i> utility on the translated pathname. |
| %f                         | The filename of the file, equivalent to the result of the <i>basename</i> utility on the translated pathname.      |
| %%                         | A '%' character.                                                                                                   |

27043           Any other '%' characters in *string* produce undefined results.

27044           If no **-o exthdr.name=string** is specified, *pax* shall use the following default  
 27045           value:

27046           **%d/PaxHeaders/%f**

27047           **globexthdr.name=string**

27048           (Applicable only to the **-x pax** format.) When used in **write** or **copy** mode  
 27049           with the appropriate options, *pax* shall create global extended header records  
 27050           with **ustar** header blocks that will be treated as regular files by previous

27051 versions of *pax*. This keyword allows user control over the name that is  
 27052 written into the **ustar** header blocks for global extended header records. The  
 27053 name shall be the contents of string, after the following character substitutions  
 27054 have been made:

| <i>string</i><br><b>Includes:</b> | <b>Replaced By:</b>                                                                                                |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------|
| %n                                | An integer that represents the sequence number of the global extended header record in the archive, starting at 1. |
| %%                                | A '%' character.                                                                                                   |

27060 Any other '%' characters in *string* produce undefined results.

27061 If no **-o globexthdr.name=***string* is specified, *pax* shall use the following  
 27062 default value:

27063 \$TMPDIR/GlobalHead.%n

27064 where \$TMPDIR represents the value of the **TMPDIR** environment variable. If  
 27065 **TMPDIR** is not set, *pax* shall use /tmp.

27066 **invalid=***action*

27067 (Applicable only to the **-x pax** format.) This keyword allows user control over  
 27068 the action *pax* takes upon encountering values in an extended header record  
 27069 that, in **read** or **copy** mode, are invalid in the destination hierarchy or, in **list**  
 27070 mode, cannot be written in the codeset and current locale of the  
 27071 implementation. The following are invalid values that shall be recognized by  
 27072 *pax*:

- 27073 — In **read** or **copy** mode, a filename or link name that contains character  
 27074 encodings invalid in the destination hierarchy. (For example, the name  
 27075 may contain embedded NULs.)
- 27076 — In **read** or **copy** mode, a filename or link name that is longer than the  
 27077 maximum allowed in the destination hierarchy (for either a pathname  
 27078 component or the entire pathname).
- 27079 — In **list** mode, any character string value (filename, link name, user name,  
 27080 and so on) that cannot be written in the codeset and current locale of the  
 27081 implementation.

27082 The following mutually-exclusive values of the *action* argument are  
 27083 supported:

27084 **bypass**      In **read** or **copy** mode, *pax* shall bypass the file, causing no  
 27085 change to the destination hierarchy. In **list** mode, *pax* shall write  
 27086 all requested valid values for the file, but its method for writing  
 27087 invalid values is unspecified.

27088 **rename**      In **read** or **copy** mode, *pax* shall act as if the **-i** option were in  
 27089 effect for each file with invalid filename or link name values,  
 27090 allowing the user to provide a replacement name interactively.  
 27091 In **list** mode, *pax* shall behave identically to the **bypass** action.

27092 **UTF-8**      When used in **read**, **copy**, or **list** mode and a filename, link  
 27093 name, owner name, or any other field in an extended header  
 27094 record cannot be translated from the **pax** UTF-8 codeset format  
 27095 to the codeset and current locale of the implementation, *pax*

|       |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27096 |                       | shall use the actual UTF-8 encoding for the name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27097 | <b>write</b>          | In <b>read</b> or <b>copy</b> mode, <i>pax</i> shall write the file, translating or truncating the name, regardless of whether this may overwrite an existing file with a valid name. In <b>list</b> mode, <i>pax</i> shall behave identically to the <b>bypass</b> action.                                                                                                                                                                                                                                                                                                                                    |
| 27098 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27099 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27100 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27101 |                       | If no <b>-o invalid=</b> option is specified, <i>pax</i> shall act as if <b>-o invalid=bypass</b> were specified. Any overwriting of existing files that may be allowed by the <b>-o invalid=</b> actions shall be subject to permission ( <b>-p</b> ) and modification time ( <b>-u</b> ) restrictions, and shall be suppressed if the <b>-k</b> option is also specified.                                                                                                                                                                                                                                    |
| 27102 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27103 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27104 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27105 | <b>linkdata</b>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27106 |                       | (Applicable only to the <b>-x pax</b> format.) In <b>write</b> mode, <i>pax</i> shall write the contents of a file to the archive even when that file is merely a hard link to a file whose contents have already been written to the archive.                                                                                                                                                                                                                                                                                                                                                                 |
| 27107 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27108 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27109 | <b>listopt=format</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27110 |                       | This keyword specifies the output format of the table of contents produced when the <b>-v</b> option is specified in <b>list</b> mode. See <b>List Mode Format Specifications</b> (on page 706). To avoid ambiguity, the <b>listopt=format</b> shall be the only or final <b>keyword=value</b> pair in a <b>-o</b> option-argument; all characters in the remainder of the option-argument shall be considered part of the format string. When multiple <b>-o listopt=format</b> options are specified, the format strings shall be considered a single, concatenated string, evaluated in command line order. |
| 27111 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27112 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27113 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27114 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27115 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27116 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27117 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27118 | <b>times</b>          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27119 |                       | (Applicable only to the <b>-x B</b> format.) When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include <b>atime</b> , <b>ctime</b> , and <b>mtime</b> extended header records for each file. See <b>pax Extended Header File Times</b> (on page 714).                                                                                                                                                                                                                                                                                                                                            |
| 27120 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27121 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27122 |                       | In addition to these keywords, if the <b>-x B</b> format is specified, any of the keywords and values defined in <b>pax Extended Header</b> (on page 711), including implementation extensions, can be used in <b>-o</b> option-arguments, in either of two modes:                                                                                                                                                                                                                                                                                                                                             |
| 27123 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27124 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27125 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27126 | <b>keyword=value</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27127 |                       | When used in <b>write</b> or <b>copy</b> mode, these keyword/value pairs shall be included at the beginning of the archive as <b>typeflag g</b> global extended header records. When used in <b>read</b> or <b>list</b> mode, these keyword/value pairs shall act as if they had been at the beginning of the archive as <b>typeflag g</b> global extended header records.                                                                                                                                                                                                                                     |
| 27128 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27129 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27130 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27131 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27132 | <b>keyword:=value</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27133 |                       | When used in <b>write</b> or <b>copy</b> mode, these keyword/value pairs shall be included as records at the beginning of a <b>typeflag x</b> extended header for each file. (This shall be equivalent to the equal-sign form except that it creates no <b>typeflag g</b> global extended header records.) When used in <b>read</b> or <b>list</b> mode, these keyword/value pairs shall act as if they were included as records at the end of each extended header; thus, they shall override any global or file-specific extended header record keywords of the same names. For example, in the command:     |
| 27134 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27135 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27136 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27137 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27138 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27139 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27140 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27141 |                       | <pre>pax -r -o " gname:=mygroup, " &lt;archive</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 27142 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27143 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

27144  
27145       the group name will be forced to a new value for all files read from the  
archive.

27146  
27147       The precedence of **-o** keywords over various fields in the archive is described in  
**pax Extended Header Keyword Precedence** (on page 714).

27148       **-p string**  
27149       Specify one or more file characteristic options (privileges). The *string* option-  
27150       argument shall be a string specifying file characteristics to be retained or discarded  
27151       on extraction. The string shall consist of the specification characters **a**, **e**, **m**, **o**, and  
27152       **p**. Other implementation-defined characters can be included. Multiple  
27153       characteristics can be concatenated within the same string and multiple **-p** options  
can be specified. The meaning of the specification characters are as follows:

- 27154       a     Do not preserve file access times.  
27155       e     Preserve the user ID, group ID, file mode bits (see the Base Definitions volume  
27156       of IEEE Std 1003.1-2001, Section 3.168, File Mode Bits), access time,  
27157       modification time, and any other implementation-defined file characteristics.  
27158       m     Do not preserve file modification times.  
27159       o     Preserve the user ID and group ID.  
27160       p     Preserve the file mode bits. Other implementation-defined file mode attributes  
27161       may be preserved.

27162       In the preceding list, “preserve” indicates that an attribute stored in the archive  
27163       shall be given to the extracted file, subject to the permissions of the invoking  
27164       process. The access and modification times of the file shall be preserved unless  
27165       otherwise specified with the **-p** option or not stored in the archive. All attributes  
27166       that are not preserved shall be determined as part of the normal file creation action  
27167       (see Section 1.7.1.4 (on page 4)).

27168       If neither the **e** nor the **o** specification character is specified, or the user ID and  
27169       group ID are not preserved for any reason, *pax* shall not set the **S\_ISUID** and  
27170       **S\_ISGID** bits of the file mode.

27171       If the preservation of any of these items fails for any reason, *pax* shall write a  
27172       diagnostic message to standard error. Failure to preserve these items shall affect  
27173       the final exit status, but shall not cause the extracted file to be deleted.

27174       If file characteristic letters in any of the *string* option-arguments are duplicated or  
27175       conflict with each other, the ones given last shall take precedence. For example, if  
27176       **-p e me** is specified, file modification times are preserved.

27177       **-s replstr**  
27178       Modify file or archive member names named by *pattern* or *file* operands according  
27179       to the substitution expression *replstr*, using the syntax of the *ed* utility. The  
27180       concepts of “address” and “line” are meaningless in the context of the *pax* utility,  
and shall not be supplied. The format shall be:

27181       **-s /old/new/**[gp]

27182       where as in *ed*, *old* is a basic regular expression and *new* can contain an ampersand,  
27183       ‘\n’ (where *n* is a digit) backreferences, or subexpression matching. The *old* string  
27184       shall also be permitted to contain <newline>s.

27185       Any non-null character can be used as a delimiter (‘/’ shown here). Multiple **-s**  
27186       expressions can be specified; the expressions shall be applied in the order  
27187       specified, terminating with the first successful substitution. The optional trailing  
27188       ‘g’ is as defined in the *ed* utility. The optional trailing ‘p’ shall cause successful

|       |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27189 |                         | substitutions to be written to standard error. File or archive member names that substitute to the empty string shall be ignored when reading and writing archives.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27190 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27191 | <b>-t</b>               | When reading files from the file system, and if the user has the permissions required by <i>utime()</i> to do so, set the access time of each file read to the access time that it had before being read by <i>pax</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27192 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27193 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27194 | <b>-u</b>               | Ignore files that are older (having a less recent file modification time) than a pre-existing file or archive member with the same name. In <b>read</b> mode, an archive member with the same name as a file in the file system shall be extracted if the archive member is newer than the file. In <b>write</b> mode, an archive file member with the same name as a file in the file system shall be superseded if the file is newer than the archive member. If <b>-a</b> is also specified, this is accomplished by appending to the archive; otherwise, it is unspecified whether this is accomplished by actual replacement in the archive or by appending to the archive. In <b>copy</b> mode, the file in the destination hierarchy shall be replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer. |
| 27195 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27196 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27197 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27198 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27199 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27200 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27201 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27202 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27203 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27204 | <b>-v</b>               | In <b>list</b> mode, produce a verbose table of contents (see the STDOUT section). Otherwise, write archive member pathnames to standard error (see the STDERR section).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27205 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27206 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27207 | <b>-x</b> <i>format</i> | Specify the output archive format. The <i>pax</i> utility shall support the following formats:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27208 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27209 | <b>cpio</b>             | The <b>cpio</b> interchange format; see the EXTENDED DESCRIPTION section. The default <i>blocksize</i> for this format for character special archive files shall be 5120. Implementations shall support all <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27210 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27211 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27212 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27213 | <b>pax</b>              | The <b>pax</b> interchange format; see the EXTENDED DESCRIPTION section. The default <i>blocksize</i> for this format for character special archive files shall be 5120. Implementations shall support all <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 27214 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27215 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27216 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27217 | <b>ustar</b>            | The <b>tar</b> interchange format; see the EXTENDED DESCRIPTION section. The default <i>blocksize</i> for this format for character special archive files shall be 10 240. Implementations shall support all <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 27218 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27219 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27220 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27221 |                         | Implementation-defined formats shall specify a default block size as well as any other block sizes supported for character special archive files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27222 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27223 |                         | Any attempt to append to an archive file in a format different from the existing archive format shall cause <i>pax</i> to exit immediately with a non-zero exit status.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 27224 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27225 |                         | In <b>copy</b> mode, if no <b>-x</b> format is specified, <i>pax</i> shall behave as if <b>-xpax</b> were specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 27226 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27227 | <b>-X</b>               | When traversing the file hierarchy specified by a pathname, <i>pax</i> shall not descend into directories that have a different device ID ( <i>st_dev</i> ; see the System Interfaces volume of IEEE Std 1003.1-2001, <i>stat()</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 27228 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27229 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27230 |                         | The options that operate on the names of files or archive members ( <b>-c</b> , <b>-i</b> , <b>-n</b> , <b>-s</b> , <b>-u</b> , and <b>-v</b> ) shall interact as follows. In <b>read</b> mode, the archive members shall be selected based on the user-specified <i>pattern</i> operands as modified by the <b>-c</b> , <b>-n</b> , and <b>-u</b> options. Then, any <b>-s</b> and <b>-i</b> options shall modify, in that order, the names of the selected files. The <b>-v</b> option shall write names resulting from these modifications.                                                                                                                                                                                                                                                                                                                                                   |
| 27231 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27232 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27233 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27234 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

27235 In **write** mode, the files shall be selected based on the user-specified pathnames as modified by  
27236 the **-n** and **-u** options. Then, any **-s** and **-i** options shall modify, in that order, the names of  
27237 these selected files. The **-v** option shall write names resulting from these modifications.

27238 If both the **-u** and **-n** options are specified, *pax* shall not consider a file selected unless it is newer  
27239 than the file to which it is compared.

27240 **List Mode Format Specifications**

27241 In **list** mode with the **-o listopt=format** option, the *format* argument shall be applied for each  
27242 selected file. The *pax* utility shall append a <newline> to the **listopt** output for each selected file.  
27243 The *format* argument shall be used as the *format* string described in the Base Definitions volume  
27244 of IEEE Std 1003.1-2001, Chapter 5, File Format Notation, with the exceptions 1. through 5.  
27245 defined in the EXTENDED DESCRIPTION section of *printf*, plus the following exceptions:

27246 6. The sequence (*keyword*) can occur before a format conversion specifier. The conversion  
27247 argument is defined by the value of *keyword*. The implementation shall support the  
27248 following keywords:

- 27249 — Any of the Field Name entries in Table 4-13 (on page 715) and Table 4-15 (on page 718).  
27250 The implementation may support the *cpio* keywords without the leading **c\_** in addition  
27251 to the form required by Table 4-16 (on page 719).
- 27252 — Any keyword defined for the extended header in **pax Extended Header** (on page 711).
- 27253 — Any keyword provided as an implementation-defined extension within the extended  
27254 header defined in **pax Extended Header** (on page 711).

27255 For example, the sequence "%(charset)s" is the string value of the name of the character  
27256 set in the extended header.

27257 The result of the keyword conversion argument shall be the value from the applicable  
27258 header field or extended header, without any trailing NULs.

27259 All keyword values used as conversion arguments shall be translated from the UTF-8  
27260 encoding to the character set appropriate for the local file system, user database, and so on,  
27261 as applicable.

27262 7. An additional conversion specifier character, T, shall be used to specify time formats. The T  
27263 conversion specifier character can be preceded by the sequence (*keyword*=*subformat*), where  
27264 *subformat* is a date format as defined by *date* operands. The default *keyword* shall be **mtime**  
27265 and the default *subformat* shall be:

27266 %b %e %H:%M %Y

27267 8. An additional conversion specifier character, M, shall be used to specify the file mode string  
27268 as defined in *ls* Standard Output. If (*keyword*) is omitted, the **mode** keyword shall be used.  
27269 For example, %.1M writes the single character corresponding to the <entry type> field of the  
27270 *ls -l* command.

27271 9. An additional conversion specifier character, D, shall be used to specify the device for block  
27272 or special files, if applicable, in an implementation-defined format. If not applicable, and  
27273 (*keyword*) is specified, then this conversion shall be equivalent to %(*keyword*)u. If not  
27274 applicable, and (*keyword*) is omitted, then this conversion shall be equivalent to <space>.

27275 10. An additional conversion specifier character, F, shall be used to specify a pathname. The F  
27276 conversion character can be preceded by a sequence of comma-separated keywords:

27277 (*keyword*[ ,*keyword*] . . . )

27278        The values for all the keywords that are non-null shall be concatenated together, each  
27279        separated by a '/'. The default shall be (**path**) if the keyword **path** is defined; otherwise,  
27280        the default shall be (**prefix.name**).

- 27281        11. An additional conversion specifier character, **L**, shall be used to specify a symbolic line  
27282        expansion. If the current file is a symbolic link, then %**L** shall expand to:

27283              "%s → %s", <value of keyword>, <contents of link>

27284        Otherwise, the %**L** conversion specification shall be the equivalent of %F.

## 27285 OPERANDS

27286        The following operands shall be supported:

27287        **directory**      The destination directory pathname for **copy** mode.

27288        **file**          A pathname of a file to be copied or archived.

27289        **pattern**       A pattern matching one or more pathnames of archive members. A pattern must  
27290        be given in the name-generating notation of the pattern matching notation in  
27291        Section 2.13 (on page 62), including the filename expansion rules in Section 2.13.3  
27292        (on page 63). The default, if no **pattern** is specified, is to select all members in the  
27293        archive.

## 27294 STDIN

27295        In **write** mode, the standard input shall be used only if no **file** operands are specified. It shall be a  
27296        text file containing a list of pathnames, one per line, without leading or trailing <blank>s.

27297        In **list** and **read** modes, if **-f** is not specified, the standard input shall be an archive file.

27298        Otherwise, the standard input shall not be used.

## 27299 INPUT FILES

27300        The input file named by the **archive** option-argument, or standard input when the archive is read  
27301        from there, shall be a file formatted according to one of the specifications in the EXTENDED  
27302        DESCRIPTION section or some other implementation-defined format.

27303        The file **/dev/tty** shall be used to write prompts and read responses.

## 27304 ENVIRONMENT VARIABLES

27305        The following environment variables shall affect the execution of **pax**:

27306        **LANG**          Provide a default value for the internationalization variables that are unset or null.  
27307              (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
27308              Internationalization Variables for the precedence of internationalization variables  
27309              used to determine the values of locale categories.)

27310        **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
27311        internationalization variables.

### 27312 **LC\_COLLATE**

27313        Determine the locale for the behavior of ranges, equivalence classes, and multi-  
27314        character collating elements used in the pattern matching expressions for the  
27315        **pattern** operand, the basic regular expression for the **-s** option, and the extended  
27316        regular expression defined for the **yesexpr** locale keyword in the **LC\_MESSAGES**  
27317        category.

27318        **LC\_CTYPE**      Determine the locale for the interpretation of sequences of bytes of text data as  
27319        characters (for example, single-byte as opposed to multi-byte characters in  
27320        arguments and input files), the behavior of character classes used in the extended  
27321        regular expression defined for the **yesexpr** locale keyword in the **LC\_MESSAGES**

|           |                            |                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27322     |                            | category, and pattern matching.                                                                                                                                                                                                                                                                                                                                                                                         |
| 27323     | <i>LC_MESSAGES</i>         | Determine the locale for the processing of affirmative responses that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                |
| 27327     | <i>LC_TIME</i>             | Determine the format and contents of date and time strings when the <b>-v</b> option is specified.                                                                                                                                                                                                                                                                                                                      |
| 27329 XSI | <i>NLSPATH</i>             | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                   |
| 27330     | <i>TMPDIR</i>              | Determine the pathname that provides part of the default global extended header record file, as described for the <b>-o globexthdr=</b> keyword in the OPTIONS section.                                                                                                                                                                                                                                                 |
| 27332     | <i>TZ</i>                  | Determine the timezone used to calculate date and time strings when the <b>-v</b> option is specified. If <i>TZ</i> is unset or null, an unspecified default timezone shall be used.                                                                                                                                                                                                                                    |
| 27334     | <b>ASYNCHRONOUS EVENTS</b> |                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27335     |                            | Default.                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27336     | <b>STDOUT</b>              |                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27337     |                            | In <b>write</b> mode, if <b>-f</b> is not specified, the standard output shall be the archive formatted according to one of the specifications in the EXTENDED DESCRIPTION section, or some other implementation-defined format (see <b>-x format</b> ).                                                                                                                                                                |
| 27340     |                            | In <b>list</b> mode, when the <b>-olistopt=format</b> has been specified, the selected archive members shall be written to standard output using the format described under <b>List Mode Format Specifications</b> (on page 706). In <b>list</b> mode without the <b>-olistopt=format</b> option, the table of contents of the selected archive members shall be written to standard output using the following format: |
| 27345     |                            | "%s\n", <pathname>                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27346     |                            | If the <b>-v</b> option is specified in <b>list</b> mode, the table of contents of the selected archive members shall be written to standard output using the following formats.                                                                                                                                                                                                                                        |
| 27348     |                            | For pathnames representing hard links to previous members of the archive:                                                                                                                                                                                                                                                                                                                                               |
| 27349     |                            | "%sΔ==Δ%s\n", <ls -l listing>, <linkname>                                                                                                                                                                                                                                                                                                                                                                               |
| 27350     |                            | For all other pathnames:                                                                                                                                                                                                                                                                                                                                                                                                |
| 27351     |                            | "%s\n", <ls -l listing>                                                                                                                                                                                                                                                                                                                                                                                                 |
| 27352     |                            | where <ls -l listing> shall be the format specified by the <i>ls</i> utility with the <b>-l</b> option. When writing pathnames in this format, it is unspecified what is written for fields for which the underlying archive format does not have the correct information, although the correct number of <blank>-separated fields shall be written.                                                                    |
| 27356     |                            | In <b>list</b> mode, standard output shall not be buffered more than a line at a time.                                                                                                                                                                                                                                                                                                                                  |
| 27357     | <b>STDERR</b>              |                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27358     |                            | If <b>-v</b> is specified in <b>read</b> , <b>write</b> , or <b>copy</b> modes, <i>pax</i> shall write the pathnames it processes to the standard error output using the following format:                                                                                                                                                                                                                              |
| 27360     |                            | "%s\n", <pathname>                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27361     |                            | These pathnames shall be written as soon as processing is begun on the file or archive member, and shall be flushed to standard error. The trailing <newline>, which shall not be buffered, is written when the file has been read or written.                                                                                                                                                                          |

27364 If the **-s** option is specified, and the replacement string has a trailing '**p**', substitutions shall be  
27365 written to standard error in the following format:

27366 "%sΔ>>Δ%s\n", <original pathname>, <new pathname>

27367 In all operating modes of **pax**, optional messages of unspecified format concerning the input  
27368 archive format and volume number, the number of files, blocks, volumes, and media parts as  
27369 well as other diagnostic messages may be written to standard error.

27370 In all formats, for both standard output and standard error, it is unspecified how non-printable  
27371 characters in pathnames or link names are written.

27372 When **pax** is in **read** mode or **list** mode, using the **-xpax** archive format, and a filename, link  
27373 name, owner name, or any other field in an extended header record cannot be translated from  
27374 the **pax** UTF-8 codeset format to the codeset and current locale of the implementation, **pax** shall  
27375 write a diagnostic message to standard error, shall process the file as described for the **-o**  
27376 **invalid**=option, and then shall process the next file in the archive.

## 27377 OUTPUT FILES

27378 In **read** mode, the extracted output files shall be of the archived file type. In **copy** mode, the  
27379 copied output files shall be the type of the file being copied. In either mode, existing files in the  
27380 destination hierarchy shall be overwritten only when all permission (**-p**), modification time (**-u**),  
27381 and invalid-value (**-oinvalid**=) tests allow it.

27382 In **write** mode, the output file named by the **-f** option-argument shall be a file formatted  
27383 according to one of the specifications in the EXTENDED DESCRIPTION section, or some other  
27384 implementation-defined format.

## 27385 EXTENDED DESCRIPTION

### 27386 pax Interchange Format

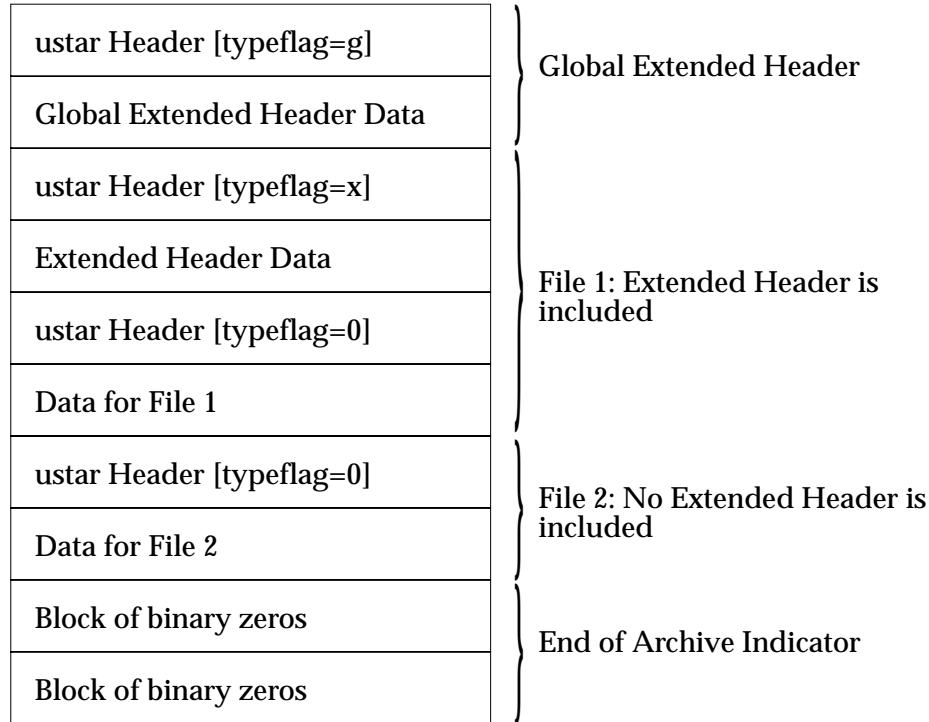
27387 A **pax** archive tape or file produced in the **-xpax** format shall contain a series of blocks. The  
27388 physical layout of the archive shall be identical to the **ustar** format described in **ustar**  
27389 **Interchange Format** (on page 714). Each file archived shall be represented by the following  
27390 sequence:

- 27391 • An optional header block with extended header records. This header block is of the form  
27392 described in **pax Header Block** (on page 710), with a *typeflag* value of **x** or **g**. The extended  
27393 header records, described in **pax Extended Header** (on page 711), shall be included as the  
27394 data for this header block.
- 27395 • A header block that describes the file. Any fields in the preceding optional extended header  
27396 shall override the associated fields in this header block for this file.
- 27397 • Zero or more blocks that contain the contents of the file.

27398 At the end of the archive file there shall be two 512-byte blocks filled with binary zeros,  
27399 interpreted as an end-of-archive indicator.

27400 A schematic of an example archive with global extended header records and two actual files is  
27401 shown in Figure 4-1 (on page 710). In the example, the second file in the archive has no extended  
27402 header preceding it, presumably because it has no need for extended attributes.

27403



27404

**Figure 4-1** pax Format Archive Example

27405

### pax Header Block

27406  
27407

The **pax** header block shall be identical to the **ustar** header block described in **ustar Interchange Format** (on page 714), except that two additional *typeflag* values are defined:

27408  
27409  
27410

- ✗ Represents extended header records for the following file in the archive (which shall have its own **ustar** header block). The format of these extended header records shall be as described in **pax Extended Header** (on page 711).

27411  
27412  
27413  
27414  
27415  
27416

- ✗ Represents global extended header records for the following files in the archive. The format of these extended header records shall be as described in **pax Extended Header** (on page 711). Each value shall affect all subsequent files that do not override that value in their own extended header record and until another global extended header record is reached that provides another value for the same field. The *typeflag g* global headers should not be used with interchange media that could suffer partial data loss in transporting the archive.

27417  
27418  
27419  
27420  
27421  
27422

For both of these types, the *size* field shall be the size of the extended header records in octets. The other fields in the header block are not meaningful to this version of the *pax* utility. However, if this archive is read by a *pax* utility conforming to the ISO POSIX-2:1993 standard, the header block fields are used to create a regular file that contains the extended header records as data. Therefore, header block field values should be selected to provide reasonable file access to this regular file.

27423  
27424  
27425

A further difference from the **ustar** header block is that data blocks for files of *typeflag 1* (the digit one) (hard link) may be included, which means that the *size* field may be greater than zero. Archives created by *pax -o linkdata* shall include these data blocks with the hard links.

27426

**pax Extended Header**

27427

A **pax** extended header contains values that are inappropriate for the **ustar** header block because of limitations in that format: fields requiring a character encoding other than that described in the ISO/IEC 646: 1991 standard, fields representing file attributes not described in the **ustar** header, and fields whose format or length do not fit the requirements of the **ustar** header. The values in an extended header add attributes to the following file (or files; see the description of the *typeflag g* header block) or override values in the following header block(s), as indicated in the following list of keywords.

27434

An extended header shall consist of one or more records, each constructed as follows:

27435

```
"%d %s=%s\n", <length>, <keyword>, <value>
```

27436

The extended header records shall be encoded according to the ISO/IEC 10646-1: 2000 standard (UTF-8). The *<length>* field, *<blank>*, equals sign, and *<newline>* shown shall be limited to the portable character set, as encoded in UTF-8. The *<keyword>* and *<value>* fields can be any UTF-8 characters. The *<length>* field shall be the decimal length of the extended header record in octets, including the trailing *<newline>*.

27437

27438

27439

27440

The *<keyword>* field shall be one of the entries from the following list or a keyword provided as an implementation extension. Keywords consisting entirely of lowercase letters, digits, and periods are reserved for future standardization. A keyword shall not include an equals sign. (In the following list, the notations “file(s)” or “block(s)” is used to acknowledge that a keyword affects the following single file after a *typeflag x* extended header, but possibly multiple files after *typeflag g*. Any requirements in the list for *pax* to include a record when in **write** or **copy** mode shall apply only when such a record has not already been provided through the use of the **-o** option. When used in **copy** mode, *pax* shall behave as if an archive had been created with applicable extended header records and then extracted.)

27441

27442

**atime** The file access time for the following file(s), equivalent to the value of the *st\_atime* member of the **stat** structure for a file, as described by the *stat()* function. The access 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 714).

27443

27444

27445

27446

27447

27448

27449

27450

27451

27452

27453

27454

27455

27456

27457

27458

27459

27460

27461

27462

27463

27464

27465

27466

27467

27468

27469

27470

27471

27472

27473

27474

27475

27476

|       | <value>                 | Formal Standard               |
|-------|-------------------------|-------------------------------|
| 27460 | ISO-IRΔ646Δ1990         | ISO/IEC 646: 1990             |
| 27461 | ISO-IRΔ8859Δ1Δ1998      | ISO/IEC 8859-1: 1998          |
| 27462 | ISO-IRΔ8859Δ2Δ1999      | ISO/IEC 8859-2: 1999          |
| 27463 | ISO-IRΔ8859Δ3Δ1999      | ISO/IEC 8859-3: 1999          |
| 27464 | ISO-IRΔ8859Δ4Δ1998      | ISO/IEC 8859-4: 1998          |
| 27465 | ISO-IRΔ8859Δ5Δ1999      | ISO/IEC 8859-5: 1999          |
| 27466 | ISO-IRΔ8859Δ6Δ1999      | ISO/IEC 8859-6: 1999          |
| 27467 | ISO-IRΔ8859Δ7Δ1987      | ISO/IEC 8859-7: 1987          |
| 27468 | ISO-IRΔ8859Δ8Δ1999      | ISO/IEC 8859-8: 1999          |
| 27469 | ISO-IRΔ8859Δ9Δ1999      | ISO/IEC 8859-9: 1999          |
| 27470 | ISO-IRΔ8859Δ10Δ1998     | ISO/IEC 8859-10: 1998         |
| 27471 | ISO-IRΔ8859Δ13Δ1998     | ISO/IEC 8859-13: 1998         |
| 27472 | ISO-IRΔ8859Δ14Δ1998     | ISO/IEC 8859-14: 1998         |
| 27473 | ISO-IRΔ8859Δ15Δ1999     | ISO/IEC 8859-15: 1999         |
| 27474 | ISO-IRΔ10646Δ2000       | ISO/IEC 10646: 2000           |
| 27475 | ISO-IRΔ10646Δ2000ΔUTF-8 | ISO/IEC 10646, UTF-8 encoding |
| 27476 | BINARY                  | None.                         |

27477

27478

27479

The encoding is included in an extended header for information only; when *pax* is used as described in IEEE Std 1003.1-2001, it shall not translate the file data into any other encoding. The **BINARY** entry indicates unencoded binary data.

27480

27481

When used in **write** or **copy** mode, it is implementation-defined whether *pax* includes a **charset** extended header record for a file.

27482

**comment**

27483

A series of characters used as a comment. All characters in the <value> field shall be ignored by *pax*.

27484

**ctime**

27485

27486

27487

27488

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 714).

27489

**gid**

27490

27491

27492

27493

The group ID of the group that owns the file, expressed as a decimal number using digits from the ISO/IEC 646: 1991 standard. This record shall override the *gid* field in the following header block(s). When used in **write** or **copy** mode, *pax* shall include a *gid* extended header record for each file whose group ID is greater than 2 097 151 (octal 7 777 777).

27494

**gname**

27495

27496

27497

27498

27499

27500

27501

27502

27503

The group of the file(s), formatted as a group name in the group database. This record shall override the *gid* and *gname* fields in the following header block(s), and any *gid* extended header record. When used in **read**, **copy**, or **list** mode, *pax* shall translate the name from the UTF-8 encoding in the header record to the character set appropriate for the group database on the receiving system. If any of the UTF-8 characters cannot be translated, and if the **-oinvalid=UTF-8** option is not specified, the results are implementation-defined. When used in **write** or **copy** mode, *pax* shall include a **gname** extended header record for each file whose group name cannot be represented entirely with the letters and digits of the portable character set.

27504

**linkpath**

27505

27506

The pathname of a link being created to another file, of any type, previously archived. This record shall override the *linkname* field in the following **ustar** header block(s). The following **ustar** header block shall determine the type of link created.

|       |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27507 |                     | If <i>typeflag</i> of the following header block is 1, it shall be a hard link. If <i>typeflag</i> is 2, it shall be a symbolic link and the <b>linkpath</b> value shall be the contents of the symbolic link. The <b>pax</b> utility shall translate the name of the link (contents of the symbolic link) from the UTF-8 encoding to the character set appropriate for the local file system. When used in <b>write</b> or <b>copy</b> mode, <b>pax</b> shall include a <b>linkpath</b> extended header record for each link whose pathname cannot be represented entirely with the members of the portable character set other than NUL.                                                                                                                                                                                                                      |
| 27514 | <b>mtime</b>        | The file modification time of the following file(s), equivalent to the value of the <i>st_mtime</i> member of the <b>stat</b> structure for a file, as described in the <i>stat()</i> function. This record shall override the <i>mtime</i> field in the following header block(s). The modification 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 <b>pax Extended Header File Times</b> (on page 714).                                                                                                                                                                                                                                                                                                                                                             |
| 27520 | <b>path</b>         | The pathname of the following file(s). This record shall override the <i>name</i> and <i>prefix</i> fields in the following header block(s). The <b>pax</b> utility shall translate the pathname of the file from the UTF-8 encoding to the character set appropriate for the local file system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27524 |                     | When used in <b>write</b> or <b>copy</b> mode, <b>pax</b> shall include a <b>path</b> extended header record for each file whose pathname cannot be represented entirely with the members of the portable character set other than NUL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27527 | <b>realtime.any</b> | The keywords prefixed by “realtime.” are reserved for future standardization.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27528 | <b>security.any</b> | The keywords prefixed by “security.” are reserved for future standardization.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27529 | <b>size</b>         | The size of the file in octets, expressed as a decimal number using digits from the ISO/IEC 646: 1991 standard. This record shall override the <i>size</i> field in the following header block(s). When used in <b>write</b> or <b>copy</b> mode, <b>pax</b> shall include a <b>size</b> extended header record for each file with a size value greater than 8 589 934 591 (octal 77 777 777 777).                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27534 | <b>uid</b>          | The user ID of the file owner, expressed as a decimal number using digits from the ISO/IEC 646: 1991 standard. This record shall override the <i>uid</i> field in the following header block(s). When used in <b>write</b> or <b>copy</b> mode, <b>pax</b> shall include a <b>uid</b> extended header record for each file whose owner ID is greater than 2 097 151 (octal 7 777 777).                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 27539 | <b>uname</b>        | The owner of the following file(s), formatted as a user name in the user database. This record shall override the <i>uid</i> and <i>uname</i> fields in the following header block(s), and any <i>uid</i> extended header record. When used in <b>read</b> , <b>copy</b> , or <b>list</b> mode, <b>pax</b> shall translate the name from the UTF-8 encoding in the header record to the character set appropriate for the user database on the receiving system. If any of the UTF-8 characters cannot be translated, and if the <b>-oinvalid=</b> UTF-8 option is not specified, the results are implementation-defined. When used in <b>write</b> or <b>copy</b> mode, <b>pax</b> shall include a <b>uname</b> extended header record for each file whose user name cannot be represented entirely with the letters and digits of the portable character set. |
| 27549 |                     | If the <value> field is zero length, it shall delete any header block field, previously entered extended header value, or global extended header value of the same name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 27551 |                     | If a keyword in an extended header record (or in a <b>-o</b> option-argument) overrides or deletes a corresponding field in the <b>ustar</b> header block, <b>pax</b> shall ignore the contents of that header block field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

27554 Unlike the **ustar** header block fields, NULs shall not delimit <value>s; all characters within the  
27555 <value> field shall be considered data for the field. None of the length limitations of the **ustar**  
27556 header block fields in Table 4-13 (on page 715) shall apply to the extended header records.

### 27557 pax Extended Header Keyword Precedence

27558 This section describes the precedence in which the various header records and fields and  
27559 command line options are selected to apply to a file in the archive. When *pax* is used in **read** or  
27560 **list** modes, it shall determine a file attribute in the following sequence:

- 27561 1. If **-odelete=keyword-prefix** is used, the affected attributes shall be determined from step 7.,  
27562 if applicable, or ignored otherwise.
- 27563 2. If **-okeyword:=** is used, the affected attributes shall be ignored.
- 27564 3. If **-okeyword:=value** is used, the affected attribute shall be assigned the value.
- 27565 4. If there is a *typeflag x* extended header record, the affected attribute shall be assigned the  
27566 <value>. When extended header records conflict, the last one given in the header shall take  
27567 precedence.
- 27568 5. If **-okeyword=value** is used, the affected attribute shall be assigned the value.
- 27569 6. If there is a *typeflag g* global extended header record, the affected attribute shall be  
27570 assigned the <value>. When global extended header records conflict, the last one given in  
27571 the global header shall take precedence.
- 27572 7. Otherwise, the attribute shall be determined from the **ustar** header block.

### 27573 pax Extended Header File Times

27574 The *pax* utility shall write an **mtime** record for each file in **write** or **copy** modes if the file's  
27575 modification time cannot be represented exactly in the **ustar** header logical record described in  
27576 **ustar Interchange Format**. This can occur if the time is out of **ustar** range, or if the file system of  
27577 the underlying implementation supports non-integer time granularities and the time is not an  
27578 integer. All of these time records shall be formatted as a decimal representation of the time in  
27579 seconds since the Epoch. If a period ('.') decimal point character is present, the digits to the  
27580 right of the point shall represent the units of a subsecond timing granularity, where the first digit  
27581 is tenths of a second and each subsequent digit is a tenth of the previous digit. In **read** or **copy**  
27582 mode, the *pax* utility shall truncate the time of a file to the greatest value that is not greater than  
27583 the input header file time. In **write** or **copy** mode, the *pax* utility shall output a time exactly if it  
27584 can be represented exactly as a decimal number, and otherwise shall generate only enough digits  
27585 so that the same time shall be recovered if the file is extracted on a system whose underlying  
27586 implementation supports the same time granularity.

### 27587 **ustar Interchange Format**

27588 A **ustar** archive tape or file shall contain a series of logical records. Each logical record shall be a  
27589 fixed-size logical record of 512 octets (see below). Although this format may be thought of as  
27590 being stored on 9-track industry-standard 12.7 mm (0.5 in) magnetic tape, other types of  
27591 transportable media are not excluded. Each file archived shall be represented by a header logical  
27592 record that describes the file, followed by zero or more logical records that give the contents of  
27593 the file. At the end of the archive file there shall be two 512-octet logical records filled with  
27594 binary zeros, interpreted as an end-of-archive indicator.

27595 The logical records may be grouped for physical I/O operations, as described under the  
27596 **-bblocksize** and **-x ustar** options. Each group of logical records may be written with a single  
27597 operation equivalent to the **write()** function. On magnetic tape, the result of this write shall be a

single tape physical block. The last physical block shall always be the full size, so logical records after the two zero logical records may contain undefined data.

27600 The header logical record shall be structured as shown in the following table. All lengths and  
27601 offsets are in decimal.

**Table 4-13** ustar Header Block

| Field Name      | Octet Offset | Length (in Octets) |
|-----------------|--------------|--------------------|
| <i>name</i>     | 0            | 100                |
| <i>mode</i>     | 100          | 8                  |
| <i>uid</i>      | 108          | 8                  |
| <i>gid</i>      | 116          | 8                  |
| <i>size</i>     | 124          | 12                 |
| <i>mtime</i>    | 136          | 12                 |
| <i>chksum</i>   | 148          | 8                  |
| <i>typeflag</i> | 156          | 1                  |
| <i>linkname</i> | 157          | 100                |
| <i>magic</i>    | 257          | 6                  |
| <i>version</i>  | 263          | 2                  |
| <i>uname</i>    | 265          | 32                 |
| <i>gname</i>    | 297          | 32                 |
| <i>devmajor</i> | 329          | 8                  |
| <i>devminor</i> | 337          | 8                  |
| <i>prefix</i>   | 345          | 155                |

All characters in the header logical record shall be represented in the coded character set of the ISO/IEC 646: 1991 standard. For maximum portability between implementations, names should be selected from characters represented by the portable filename character set as octets with the most significant bit zero. If an implementation supports the use of characters outside of slash and the portable filename character set in names for files, users, and groups, one or more implementation-defined encodings of these characters shall be provided for interchange purposes.

However, the *pax* utility shall never create filenames on the local system that cannot be accessed via the procedures described in IEEE Std 1003.1-2001. If a filename is found on the medium that would create an invalid filename, it is implementation-defined whether the data from the file is stored on the file hierarchy and under what name it is stored. The *pax* utility may choose to ignore these files as long as it produces an error indicating that the file is being ignored.

27632 Each field within the header logical record is contiguous; that is, there is no padding used. Each  
27633 character on the archive medium shall be stored contiguously.

The fields *magic*, *uname*, and *gname* are character strings each terminated by a NUL character. The fields *name*, *linkname*, and *prefix* are NUL-terminated character strings except when all characters in the array contain non-NUL characters including the last character. The *version* field is two octets containing the characters "00" (zero-zero). The *typeflag* contains a single character. All other fields are leading zero-filled octal numbers using digits from the ISO/IEC 646: 1991 standard IRV. Each numeric field is terminated by one or more <space> or NUL characters.

The *name* and the *prefix* fields shall produce the pathname of the file. A new pathname shall be formed, if *prefix* is not an empty string (its first character is not NUL), by concatenating *prefix* (up to the first NUL character), a slash character, and *name*; otherwise, *name* is used alone. In either case, *name* is terminated at the first NUL character. If *prefix* begins with a NUL character, it shall be ignored. In this manner, pathnames of at most 256 characters can be supported. If a pathname

27645 does not fit in the space provided, *pax* shall notify the user of the error, and shall not store any  
27646 part of the file—header or data—on the medium.

27647 The *linkname* field, described below, shall not use the *prefix* to produce a pathname. As such, a  
27648 *linkname* is limited to 100 characters. If the name does not fit in the space provided, *pax* shall  
27649 notify the user of the error, and shall not attempt to store the link on the medium.

27650 The *mode* field provides 12 bits encoded in the ISO/IEC 646:1991 standard octal digit representation. The encoded bits shall represent the following values:

**Table 4-14** ustar mode Field

| <b>Bit Value</b> | <b>IEEE Std 1003.1-2001 Bit</b> | <b>Description</b>                              |
|------------------|---------------------------------|-------------------------------------------------|
| 04 000           | S_ISUID                         | Set UID on execution.                           |
| 02 000           | S_ISGID                         | Set GID on execution.                           |
| 01 000           | <reserved>                      | Reserved for future standardization.            |
| 00 400           | S_IRUSR                         | Read permission for file owner class.           |
| 00 200           | S_IWUSR                         | Write permission for file owner class.          |
| 00 100           | S_IXUSR                         | Execute/search permission for file owner class. |
| 00 040           | S_IRGRP                         | Read permission for file group class.           |
| 00 020           | S_IWGRP                         | Write permission for file group class.          |
| 00 010           | S_IXGRP                         | Execute/search permission for file group class. |
| 00 004           | S_IROTH                         | Read permission for file other class.           |
| 00 002           | S_IWOTH                         | Write permission for file other class.          |
| 00 001           | S_IXOTH                         | Execute/search permission for file other class. |

When appropriate privilege is required to set one of these mode bits, and the user restoring the files from the archive does not have the appropriate privilege, the mode bits for which the user does not have appropriate privilege shall be ignored. Some of the mode bits in the archive format are not mentioned elsewhere in this volume of IEEE Std 1003.1-2001. If the implementation does not support those bits, they may be ignored.

The *uid* and *gid* fields are the user and group ID of the owner and group of the file, respectively.

The *size* field is the size of the file in octets. If the *typeflag* field is set to specify a file to be of type 1 (a link) or 2 (a symbolic link), the *size* field shall be specified as zero. If the *typeflag* field is set to specify a file of type 5 (directory), the *size* field shall be interpreted as described under the definition of that record type. No data logical records are stored for types 1, 2, or 5. If the *typeflag* field is set to 3 (character special file), 4 (block special file), or 6 (FIFO), the meaning of the *size* field is unspecified by this volume of IEEE Std 1003.1-2001, and no data logical records shall be stored on the medium. Additionally, for type 6, the *size* field shall be ignored when reading. If the *typeflag* field is set to any other value, the number of logical records written following the header shall be  $(\text{size}+511)/512$ , ignoring any fraction in the result of the division.

The *mtime* field shall be the modification time of the file at the time it was archived. It is the ISO/IEC 646: 1991 standard representation of the octal value of the modification time obtained from the *stat()* function.

The *chksm* field shall be the ISO/IEC 646:1991 standard IRV representation of the octal value of the simple sum of all octets in the header logical record. Each octet in the header shall be treated as an unsigned value. These values shall be added to an unsigned integer, initialized to zero, the precision of which is not less than 17 bits. When calculating the checksum, the *chksm* field is treated as if it were all spaces.

The *typeflag* field specifies the type of file archived. If a particular implementation does not recognize the type, or the user does not have appropriate privilege to create that type, the file

shall be extracted as if it were a regular file if the file type is defined to have a meaning for the *size* field that could cause data logical records to be written on the medium (see the previous description for *size*). If conversion to a regular file occurs, the *pax* utility shall produce an error indicating that the conversion took place. All of the *typeflag* fields shall be coded in the ISO/IEC 646: 1991 standard IRV:

- 27691 0 Represents a regular file. For backwards-compatibility, a *typeflag* value of binary zero ('`\0`') should be recognized as meaning a regular file when extracting files from the 27692 archive. Archives written with this version of the archive file format create regular files 27693 with a *typeflag* value of the ISO/IEC 646: 1991 standard IRV '`0`'.
- 27694 1 Represents a file linked to another file, of any type, previously archived. Such files are 27695 identified by each file having the same device and file serial number. The linked-to 27696 name is specified in the *linkname* field with a NUL-character terminator if it is less than 27697 100 octets in length.
- 27698 2 Represents a symbolic link. The contents of the symbolic link shall be stored in the 27699 *linkname* field.
- 27700 3 , 4 Represent character special files and block special files respectively. In this case the 27701 *devmajor* and *devminor* fields shall contain information defining the device, the format 27702 of which is unspecified by this volume of IEEE Std 1003.1-2001. Implementations may 27703 map the device specifications to their own local specification or may ignore the entry.
- 27704 5 Specifies a directory or subdirectory. On systems where disk allocation is performed on 27705 a directory basis, the *size* field shall contain the maximum number of octets (which may 27706 be rounded to the nearest disk block allocation unit) that the directory may hold. A *size* 27707 field of zero indicates no such limiting. Systems that do not support limiting in this 27708 manner should ignore the *size* field.
- 27709 6 Specifies a FIFO special file. Note that the archiving of a FIFO file archives the existence 27710 of this file and not its contents.
- 27711 7 Reserved to represent a file to which an implementation has associated some high- 27712 performance attribute. Implementations without such extensions should treat this file 27713 as a regular file (type 0).
- 27714 A-Z The letters '`A`' to '`Z`', inclusive, are reserved for custom implementations. All other 27715 values are reserved for future versions of IEEE Std 1003.1-2001.

Attempts to archive a socket using **ustar** interchange format shall produce a diagnostic message. Handling of other file types is implementation-defined.

The *magic* field is the specification that this archive was output in this archive format. If this field contains **ustar** (the five characters from the ISO/IEC 646: 1991 standard IRV shown followed by NUL), the *uname* and *gname* fields shall contain the ISO/IEC 646: 1991 standard IRV representation of the owner and group of the file, respectively (truncated to fit, if necessary). When the file is restored by a privileged, protection-preserving version of the utility, the user and group databases shall be scanned for these names. If found, the user and group IDs contained within these files shall be used rather than the values contained within the *uid* and *gid* fields.

27732      **cpio Interchange Format**

27733      The octet-oriented **cpio** archive format shall be a series of entries, each comprising a header that  
 27734      describes the file, the name of the file, and then the contents of the file.

27735      An archive may be recorded as a series of fixed-size blocks of octets. This blocking shall be used  
 27736      only to make physical I/O more efficient. The last group of blocks shall always be at the full  
 27737      size.

27738      For the octet-oriented **cpio** archive format, the individual entry information shall be in the order  
 27739      indicated and described by the following table; see also the <**cpio.h**> header.

27740      **Table 4-15 Octet-Oriented cpio Archive Entry**

| Header Field Name    | Length (in Octets) | Interpreted as  |
|----------------------|--------------------|-----------------|
| <i>c_magic</i>       | 6                  | Octal number    |
| <i>c_dev</i>         | 6                  | Octal number    |
| <i>c_ino</i>         | 6                  | Octal number    |
| <i>c_mode</i>        | 6                  | Octal number    |
| <i>c_uid</i>         | 6                  | Octal number    |
| <i>c_gid</i>         | 6                  | Octal number    |
| <i>c_nlink</i>       | 6                  | Octal number    |
| <i>c_rdev</i>        | 6                  | Octal number    |
| <i>c_mtime</i>       | 11                 | Octal number    |
| <i>c_namesize</i>    | 6                  | Octal number    |
| <i>c_filesize</i>    | 11                 | Octal number    |
| Filename Field Name  | Length             | Interpreted as  |
| <i>c_name</i>        | <i>c_namesize</i>  | Pathname string |
| File Data Field Name | Length             | Interpreted as  |
| <i>c_filedata</i>    | <i>c_filesize</i>  | Data            |

27757      **cpio Header**

27758      For each file in the archive, a header as defined previously shall be written. The information in  
 27759      the header fields is written as streams of the ISO/IEC 646: 1991 standard characters interpreted  
 27760      as octal numbers. The octal numbers shall be extended to the necessary length by appending the  
 27761      ISO/IEC 646: 1991 standard IRV zeros at the most-significant-digit end of the number; the result  
 27762      is written to the most-significant digit of the stream of octets first. The fields shall be interpreted  
 27763      as follows:

- 27764      *c\_magic*      Identify the archive as being a transportable archive by containing the identifying  
                               value "070707".
- 27765      *c\_dev*, *c\_ino*   Contains values that uniquely identify the file within the archive (that is, no files  
                               contain the same pair of *c\_dev* and *c\_ino* values unless they are links to the same  
                               file). The values shall be determined in an unspecified manner.
- 27766      *c\_mode*      Contains the file type and access permissions as defined in the following table.

27770

**Table 4-16** Values for cpio c\_mode Field

| 27771 | <b>File Permissions Name</b> | <b>Value</b> | <b>Indicates</b>       |
|-------|------------------------------|--------------|------------------------|
| 27772 | C_IRUSR                      | 000 400      | Read by owner          |
| 27773 | C_IWUSR                      | 000 200      | Write by owner         |
| 27774 | C_IXUSR                      | 000 100      | Execute by owner       |
| 27775 | C_IRGRP                      | 000 040      | Read by group          |
| 27776 | C_IWGRP                      | 000 020      | Write by group         |
| 27777 | C_IXGRP                      | 000 010      | Execute by group       |
| 27778 | C_IROTH                      | 000 004      | Read by others         |
| 27779 | C_IWOTH                      | 000 002      | Write by others        |
| 27780 | C_IXOTH                      | 000 001      | Execute by others      |
| 27781 | C_ISUID                      | 004 000      | Set <i>uid</i>         |
| 27782 | C_ISGID                      | 002 000      | Set <i>gid</i>         |
| 27783 | C_ISVTX                      | 001 000      | Reserved               |
| 27784 | <b>File Type Name</b>        | <b>Value</b> | <b>Indicates</b>       |
| 27785 | C_ISDIR                      | 040 000      | Directory              |
| 27786 | C_ISFIFO                     | 010 000      | FIFO                   |
| 27787 | C_ISREG                      | 0100 000     | Regular file           |
| 27788 | C_ISLNK                      | 0120 000     | Symbolic link          |
| 27789 | C_ISBLK                      | 060 000      | Block special file     |
| 27790 | C_ISCHR                      | 020 000      | Character special file |
| 27791 | C_ISSOCK                     | 0140 000     | Socket                 |
| 27792 | C_ISCTG                      | 0110 000     | Reserved               |

27793 Directories, FIFOs, symbolic links, and regular files shall be supported on a system conforming to this volume of IEEE Std 1003.1-2001; additional values defined previously are reserved for compatibility with existing systems. Additional file types may be supported; however, such files should not be written to archives intended to be transported to other systems.

|       |                   |                                                                                                                         |
|-------|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| 27798 | <i>c_uid</i>      | Contains the user ID of the owner.                                                                                      |
| 27799 | <i>c_gid</i>      | Contains the group ID of the group.                                                                                     |
| 27800 | <i>c_nlink</i>    | Contains the number of links referencing the file at the time the archive was created.                                  |
| 27801 |                   |                                                                                                                         |
| 27802 | <i>c_rdev</i>     | Contains implementation-defined information for character or block special files.                                       |
| 27803 | <i>c_mtime</i>    | Contains the latest time of modification of the file at the time the archive was created.                               |
| 27804 |                   |                                                                                                                         |
| 27805 | <i>c_namesize</i> | Contains the length of the pathname, including the terminating NUL character.                                           |
| 27806 | <i>c_filesize</i> | Contains the length of the file in octets. This shall be the length of the data section following the header structure. |
| 27807 |                   |                                                                                                                         |

27808       **cpio Filename**

27809       The *c\_name* field shall contain the pathname of the file. The length of this field in octets is the  
27810       value of *c\_namesize*.

27811       If a filename is found on the medium that would create an invalid pathname, it is  
27812       implementation-defined whether the data from the file is stored on the file hierarchy and under  
27813       what name it is stored.

27814       All characters shall be represented in the ISO/IEC 646:1991 standard IRV. For maximum  
27815       portability between implementations, names should be selected from characters represented by  
27816       the portable filename character set as octets with the most significant bit zero. If an  
27817       implementation supports the use of characters outside the portable filename character set in  
27818       names for files, users, and groups, one or more implementation-defined encodings of these  
27819       characters shall be provided for interchange purposes. However, the *pax* utility shall never  
27820       create filenames on the local system that cannot be accessed via the procedures described  
27821       previously in this volume of IEEE Std 1003.1-2001. If a filename is found on the medium that  
27822       would create an invalid filename, it is implementation-defined whether the data from the file is  
27823       stored on the local file system and under what name it is stored. The *pax* utility may choose to  
27824       ignore these files as long as it produces an error indicating that the file is being ignored.

27825       **cpio File Data**

27826       Following *c\_name*, there shall be *c\_filesize* octets of data. Interpretation of such data occurs in a  
27827       manner dependent on the file. If *c\_filesize* is zero, no data shall be contained in *c\_filedata*.

27828       When restoring from an archive:

- If the user does not have the appropriate privilege to create a file of the specified type, *pax*  
27830       shall ignore the entry and write an error message to standard error.
- Only regular files have data to be restored. Presuming a regular file meets any selection  
27832       criteria that might be imposed on the format-reading utility by the user, such data shall be  
27833       restored.
- If a user does not have appropriate privilege to set a particular mode flag, the flag shall be  
27835       ignored. Some of the mode flags in the archive format are not mentioned elsewhere in this  
27836       volume of IEEE Std 1003.1-2001. If the implementation does not support those flags, they  
27837       may be ignored.

27838       **cpio Special Entries**

27839       FIFO special files, directories, and the trailer shall be recorded with *c\_filesize* equal to zero. For  
27840       other special files, *c\_filesize* is unspecified by this volume of IEEE Std 1003.1-2001. The header for  
27841       the next file entry in the archive shall be written directly after the last octet of the file entry  
27842       preceding it. A header denoting the filename **TRAILER!!!** shall indicate the end of the archive;  
27843       the contents of octets in the last block of the archive following such a header are undefined.

27844       **EXIT STATUS**

27845       The following exit values shall be returned:

- 0     All files were processed successfully.
- >0    An error occurred.

## 27848 CONSEQUENCES OF ERRORS

27849 If *pax* cannot create a file or a link when reading an archive or cannot find a file when writing an  
27850 archive, or cannot preserve the user ID, group ID, or file mode when the **-p** option is specified, a  
27851 diagnostic message shall be written to standard error and a non-zero exit status shall be  
27852 returned, but processing shall continue. In the case where *pax* cannot create a link to a file, *pax*  
27853 shall not, by default, create a second copy of the file.

27854 If the extraction of a file from an archive is prematurely terminated by a signal or error, *pax* may  
27855 have only partially extracted the file or (if the **-n** option was not specified) may have extracted a  
27856 file of the same name as that specified by the user, but which is not the file the user wanted.  
27857 Additionally, the file modes of extracted directories may have additional bits from the S\_IRWXU  
27858 mask set as well as incorrect modification and access times.

## 27859 APPLICATION USAGE

27860 The **-p** (privileges) option was invented to reconcile differences between historical *tar* and *cpio*  
27861 implementations. In particular, the two utilities use **-m** in diametrically opposed ways. The **-p**  
27862 option also provides a consistent means of extending the ways in which future file attributes can  
27863 be addressed, such as for enhanced security systems or high-performance files. Although it may  
27864 seem complex, there are really two modes that are most commonly used:

27865 **-p e** “Preserve everything”. This would be used by the historical superuser, someone with  
27866 all the appropriate privileges, to preserve all aspects of the files as they are recorded in  
27867 the archive. The **e** flag is the sum of **o** and **p**, and other implementation-defined  
27868 attributes.

27869 **-p p** “Preserve” the file mode bits. This would be used by the user with regular privileges  
27870 who wished to preserve aspects of the file other than the ownership. The file times are  
27871 preserved by default, but two other flags are offered to disable these and use the time  
27872 of extraction.

27873 The one pathname per line format of standard input precludes pathnames containing  
27874 <newline>s. Although such pathnames violate the portable filename guidelines, they may exist  
27875 and their presence may inhibit usage of *pax* within shell scripts. This problem is inherited from  
27876 historical archive programs. The problem can be avoided by listing filename arguments on the  
27877 command line instead of on standard input.

27878 It is almost certain that appropriate privileges are required for *pax* to accomplish parts of this  
27879 volume of IEEE Std 1003.1-2001. Specifically, creating files of type block special or character  
27880 special, restoring file access times unless the files are owned by the user (the **-t** option), or  
27881 preserving file owner, group, and mode (the **-p** option) all probably require appropriate  
27882 privileges.

27883 In **read** mode, implementations are permitted to overwrite files when the archive has multiple  
27884 members with the same name. This may fail if permissions on the first version of the file do not  
27885 permit it to be overwritten.

27886 The **cpio** and **ustar** formats can only support files up to 8 589 934 592 bytes ( $8 * 2^{30}$ ) in size.

## 27887 EXAMPLES

27888 The following command:

27889 `pax -w -f /dev/rmt/1m .`

27890 copies the contents of the current directory to tape drive 1, medium density (assuming historical  
27891 System V device naming procedures—the historical BSD device name would be `/dev/rmt9`).

27892 The following commands:

```
27893 mkdir newdir
27894 pax -rw olddir newdir
27895 copy the olddir directory hierarchy to newdir.
27896 pax -r -s ',^///*usr//*,,,' -f a.pax
27897 reads the archive a.pax, with all files rooted in /usr in the archive extracted relative to the current
27898 directory.

27899 Using the option:
27900 -o listopt="%M %(atime)T %(size)D %(name)s"
27901 overrides the default output description in Standard Output and instead writes:
27902 -rw-rw--- Jan 12 15:53 1492 /usr/foo/bar
27903 Using the options:
27904 -o listopt='%L\t%(size)D\n%.7' \
27905 -o listopt='%(name)s\n%(ctime)T\n%T'
27906 overrides the default output description in Standard Output and instead writes:
27907 /usr/foo/bar -> /tmp 1492
27908 /usr/fo
27909 Jan 12 1991
27910 Jan 31 15:53
```

## 27911 RATIONALE

27912 The *pax* utility was new for the ISO POSIX-2:1993 standard. It represents a peaceful compromise
27913 between advocates of the historical *tar* and *cpio* utilities.

27914 A fundamental difference between *cpio* and *tar* was in the way directories were treated. The *cpio*
27915 utility did not treat directories differently from other files, and to select a directory and its
27916 contents required that each file in the hierarchy be explicitly specified. For *tar*, a directory
27917 matched every file in the file hierarchy it rooted.

27918 The *pax* utility offers both interfaces; by default, directories map into the file hierarchy they root.
27919 The **-d** option causes *pax* to skip any file not explicitly referenced, as *cpio* historically did. The *tar*
27920 *-style* behavior was chosen as the default because it was believed that this was the more
27921 common usage and because *tar* is the more commonly available interface, as it was historically
27922 provided on both System V and BSD implementations.

27923 The data interchange format specification in this volume of IEEE Std 1003.1-2001 requires that
27924 processes with “appropriate privileges” shall always restore the ownership and permissions of
27925 extracted files exactly as archived. If viewed from the historic equivalence between superuser
27926 and “appropriate privileges”, there are two problems with this requirement. First, users running
27927 as superusers may unknowingly set dangerous permissions on extracted files. Second, it is
27928 needlessly limiting, in that superusers cannot extract files and own them as superuser unless the
27929 archive was created by the superuser. (It should be noted that restoration of ownerships and
27930 permissions for the superuser, by default, is historical practice in *cpio*, but not in *tar*.) In order to
27931 avoid these two problems, the *pax* specification has an additional “privilege” mechanism, the **-p**
27932 option. Only a *pax* invocation with the privileges needed, and which has the **-p** option set using
27933 the **e** specification character, has the “appropriate privilege” to restore full ownership and
27934 permission information.

27935 Note also that this volume of IEEE Std 1003.1-2001 requires that the file ownership and access
27936 permissions shall be set, on extraction, in the same fashion as the *creat()* function when provided

27937 with the mode stored in the archive. This means that the file creation mask of the user is applied  
27938 to the file permissions.

27939 Users should note that directories may be created by *pax* while extracting files with permissions  
27940 that are different from those that existed at the time the archive was created. When extracting  
27941 sensitive information into a directory hierarchy that no longer exists, users are encouraged to set  
27942 their file creation mask appropriately to protect these files during extraction.

27943 The table of contents output is written to standard output to facilitate pipeline processing.

27944 An early proposal had hard links displaying for all pathnames. This was removed because it  
27945 complicates the output of the case where **-v** is not specified and does not match historical *cpio*  
27946 usage. The hard-link information is available in the **-v** display.

27947 The description of the **-l** option allows implementations to make hard links to symbolic links.  
27948 IEEE Std 1003.1-2001 does not specify any way to create a hard link to a symbolic link, but many  
27949 implementations provide this capability as an extension. If there are hard links to symbolic links  
27950 when an archive is created, the implementation is required to archive the hard link in the archive  
27951 (unless **-H** or **-L** is specified). When in **read** mode and in **copy** mode, implementations  
27952 supporting hard links to symbolic links should use them when appropriate.

27953 The archive formats inherited from the POSIX.1-1990 standard have certain restrictions that  
27954 have been brought along from historical usage. For example, there are restrictions on the length  
27955 of pathnames stored in the archive. When *pax* is used in **copy(-rw)** mode (copying directory  
27956 hierarchies), the ability to use extensions from the **-xpax** format overcomes these restrictions.

27957 The default *blocksize* value of 5 120 bytes for *cpio* was selected because it is one of the standard  
27958 block-size values for *cpio*, set when the **-B** option is specified. (The other default block-size value  
27959 for *cpio* is 512 bytes, and this was considered to be too small.) The default block value of 10 240  
27960 bytes for *tar* was selected because that is the standard block-size value for BSD *tar*. The  
27961 maximum block size of 32 256 bytes ( $2^{15}$ –512 bytes) is the largest multiple of 512 bytes that fits  
27962 into a signed 16-bit tape controller transfer register. There are known limitations in some  
27963 historical systems that would prevent larger blocks from being accepted. Historical values were  
27964 chosen to improve compatibility with historical scripts using *dd* or similar utilities to manipulate  
27965 archives. Also, default block sizes for any file type other than character special file has been  
27966 deleted from this volume of IEEE Std 1003.1-2001 as unimportant and not likely to affect the  
27967 structure of the resulting archive.

27968 Implementations are permitted to modify the block-size value based on the archive format or  
27969 the device to which the archive is being written. This is to provide implementations with the  
27970 opportunity to take advantage of special types of devices, and it should not be used without a  
27971 great deal of consideration as it almost certainly decreases archive portability.

27972 The intended use of the **-n** option was to permit extraction of one or more files from the archive  
27973 without processing the entire archive. This was viewed by the standard developers as offering  
27974 significant performance advantages over historical implementations. The **-n** option in early  
27975 proposals had three effects; the first was to cause special characters in patterns to not be treated  
27976 specially. The second was to cause only the first file that matched a pattern to be extracted. The  
27977 third was to cause *pax* to write a diagnostic message to standard error when no file was found  
27978 matching a specified pattern. Only the second behavior is retained by this volume of  
27979 IEEE Std 1003.1-2001, for many reasons. First, it is in general not acceptable for a single option to  
27980 have multiple effects. Second, the ability to make pattern matching characters act as normal  
27981 characters is useful for parts of *pax* other than file extraction. Third, a finer degree of control over  
27982 the special characters is useful because users may wish to normalize only a single special  
27983 character in a single filename. Fourth, given a more general escape mechanism, the previous  
27984 behavior of the **-n** option can be easily obtained using the **-s** option or a *sed* script. Finally,

27985 writing a diagnostic message when a pattern specified by the user is unmatched by any file is  
27986 useful behavior in all cases.

27987 In this version, the **-n** was removed from the **copy** mode synopsis of *pax*; it is inapplicable  
27988 because there are no pattern operands specified in this mode.

27989 There is another method than *pax* for copying subtrees in IEEE Std 1003.1-2001 described as part  
27990 of the *cp* utility. Both methods are historical practice: *cp* provides a simpler, more intuitive  
27991 interface, while *pax* offers a finer granularity of control. Each provides additional functionality to  
27992 the other; in particular, *pax* maintains the hard-link structure of the hierarchy while *cp* does not.  
27993 It is the intention of the standard developers that the results be similar (using appropriate option  
27994 combinations in both utilities). The results are not required to be identical; there seemed  
27995 insufficient gain to applications to balance the difficulty of implementations having to guarantee  
27996 that the results would be exactly identical.

27997 A single archive may span more than one file. It is suggested that implementations provide  
27998 informative messages to the user on standard error whenever the archive file is changed.

27999 The **-d** option (do not create intermediate directories not listed in the archive) found in early  
28000 proposals was originally provided as a complement to the historic **-d** option of *cpio*. It has been  
28001 deleted.

28002 The **-s** option in early proposals specified a subset of the substitution command from the *ed*  
28003 utility. As there was no reason for only a subset to be supported, the **-s** option is now  
28004 compatible with the current *ed* specification. Since the delimiter can be any non-null character,  
28005 the following usage with single spaces is valid:

28006 `pax -s " foo bar " ...`

28007 The **-t** description is worded so as to note that this may cause the access time update caused by  
28008 some other activity (which occurs while the file is being read) to be overwritten.

28009 The default behavior of *pax* with regard to file modification times is the same as historical  
28010 implementations of *tar*. It is not the historical behavior of *cpio*.

28011 Because the **-i** option uses */dev/tty*, utilities without a controlling terminal are not able to use  
28012 this option.

28013 The **-y** option, found in early proposals, has been deleted because a line containing a single  
28014 period for the **-i** option has equivalent functionality. The special lines for the **-i** option (a single  
28015 period and the empty line) are historical practice in *cpio*.

28016 In early drafts, a **-echarmap** option was included to increase portability of files between systems  
28017 using different coded character sets. This option was omitted because it was apparent that  
28018 consensus could not be formed for it. In this version, the use of UTF-8 should be an adequate  
28019 substitute.

28020 The **-k** option was added to address international concerns about the dangers involved in the  
28021 character set transformations of **-e** (if the target character set were different from the source, the  
28022 filenames might be transformed into names matching existing files) and also was made more  
28023 general to protect files transferred between file systems with different {NAME\_MAX} values  
28024 (truncating a filename on a smaller system might also inadvertently overwrite existing files). As  
28025 stated, it prevents any overwriting, even if the target file is older than the source. This version  
28026 adds more granularity of options to solve this problem by introducing the **-oinvalid=** option—  
28027 specifically the UTF-8 action. (Note that an existing file that is named with a UTF-8 encoding is  
28028 still subject to overwriting in this case. The **-k** option closes that loophole.)

28029 Some of the file characteristics referenced in this volume of IEEE Std 1003.1-2001 might not be  
28030 supported by some archive formats. For example, neither the **tar** nor **cpio** formats contain the

28031 file access time. For this reason, the `e` specification character has been provided, intended to  
28032 cause all file characteristics specified in the archive to be retained.

28033 It is required that extracted directories, by default, have their access and modification times and  
28034 permissions set to the values specified in the archive. This has obvious problems in that the  
28035 directories are almost certainly modified after being extracted and that directory permissions  
28036 may not permit file creation. One possible solution is to create directories with the mode  
28037 specified in the archive, as modified by the *umask* of the user, with sufficient permissions to  
28038 allow file creation. After all files have been extracted, *pax* would then reset the access and  
28039 modification times and permissions as necessary.

28040 The list-mode formatting description borrows heavily from the one defined by the *printf* utility.  
28041 However, since there is no separate operand list to get conversion arguments, the format was  
28042 extended to allow specifying the name of the conversion argument as part of the conversion  
28043 specification.

28044 The `T` conversion specifier allows time fields to be displayed in any of the date formats. Unlike  
28045 the *ls* utility, *pax* does not adjust the format when the date is less than six months in the past.  
28046 This makes parsing the output more predictable.

28047 The `D` conversion specifier handles the ability to display the major/minor or file size, as with *ls*,  
28048 by using `%-8(size)D`.

28049 The `L` conversion specifier handles the *ls* display for symbolic links.

28050 Conversion specifiers were added to generate existing known types used for *ls*.

### 28051 pax Interchange Format

28052 The new POSIX data interchange format was developed primarily to satisfy international  
28053 concerns that the **ustar** and **cpio** formats did not provide for file, user, and group names encoded  
28054 in characters outside a subset of the ISO/IEC 646:1991 standard. The standard developers  
28055 realized that this new POSIX data interchange format should be very extensible because there  
28056 were other requirements they foresaw in the near future:

- 28057 • Support international character encodings and locale information
- 28058 • Support security information (ACLs, and so on)
- 28059 • Support future file types, such as realtime or contiguous files
- 28060 • Include data areas for implementation use
- 28061 • Support systems with words larger than 32 bits and timers with subsecond granularity

28062 The following were not goals for this format because these are better handled by separate  
28063 utilities or are inappropriate for a portable format:

- 28064 • Encryption
- 28065 • Compression
- 28066 • Data translation between locales and codesets
- 28067 • *inode* storage

28068 The format chosen to support the goals is an extension of the **ustar** format. Of the two formats  
28069 previously available, only the **ustar** format was selected for extensions because:

- 28070 • It was easier to extend in an upwards-compatible way. It offered version flags and header  
28071 block type fields with room for future standardization. The **cpio** format, while possessing a  
28072 more flexible file naming methodology, could not be extended without breaking some

28073 theoretical implementation or using a dummy filename that could be a legitimate filename.

28074 • Industry experience since the original “tar wars” fought in developing the ISO POSIX-1  
28075 standard has clearly been in favor of the **ustar** format, which is generally the default output  
28076 format selected for *pax* implementations on new systems.

28077 The new format was designed with one additional goal in mind: reasonable behavior when an  
28078 older *tar* or *pax* utility happened to read an archive. Since the POSIX.1-1990 standard mandated  
28079 that a “format-reading utility” had to treat unrecognized *typeflag* values as regular files, this  
28080 allowed the format to include all the extended information in a pseudo-regular file that preceded  
28081 each real file. An option is given that allows the archive creator to set up reasonable names for  
28082 these files on the older systems. Also, the normative text suggests that reasonable file access  
28083 values be used for this **ustar** header block. Making these header files inaccessible for convenient  
28084 reading and deleting would not be reasonable. File permissions of 600 or 700 are suggested.

28085 The **ustar** *typeflag* field was used to accommodate the additional functionality of the new format  
28086 rather than magic or version because the POSIX.1-1990 standard (and, by reference, the previous  
28087 version of *pax*), mandated the behavior of the format-reading utility when it encountered an  
28088 unknown *typeflag*, but was silent about the other two fields.

28089 Early proposals of the first revision to IEEE Std 1003.1-2001 contained a proposed archive format  
28090 that was based on compatibility with the standard for tape files (ISO 1001, similar to the format  
28091 used historically on many mainframes and minicomputers). This format was overly complex  
28092 and required considerable overhead in volume and header records. Furthermore, the standard  
28093 developers felt that it would not be acceptable to the community of POSIX developers, so it was  
28094 later changed to be a format more closely related to historical practice on POSIX systems.

28095 The prefix and name split of pathnames in **ustar** was replaced by the single path extended  
28096 header record for simplicity.

28097 The concept of a global extended header (*typeflagg*) was controversial. If this were applied to an  
28098 archive being recorded on magnetic tape, a few unreadable blocks at the beginning of the tape  
28099 could be a serious problem; a utility attempting to extract as many files as possible from a  
28100 damaged archive could lose a large percentage of file header information in this case. However,  
28101 if the archive were on a reliable medium, such as a CD-ROM, the global extended header offers  
28102 considerable potential size reductions by eliminating redundant information. Thus, the text  
28103 warns against using the global method for unreliable media and provides a method for  
28104 implanting global information in the extended header for each file, rather than in the *typeflag g*  
28105 records.

28106 No facility for data translation or filtering on a per-file basis is included because the standard  
28107 developers could not invent an interface that would allow this in an efficient manner. If a filter,  
28108 such as encryption or compression, is to be applied to all the files, it is more efficient to apply the  
28109 filter to the entire archive as a single file. The standard developers considered interfaces that  
28110 would invoke a shell script for each file going into or out of the archive, but the system overhead  
28111 in this approach was considered to be too high.

28112 One such approach would be to have **filter=** records that give a pathname for an executable.  
28113 When the program is invoked, the file and archive would be open for standard input/output  
28114 and all the header fields would be available as environment variables or command-line  
28115 arguments. The standard developers did discuss such schemes, but they were omitted from  
28116 IEEE Std 1003.1-2001 due to concerns about excessive overhead. Also, the program itself would  
28117 need to be in the archive if it were to be used portably.

28118 There is currently no portable means of identifying the character set(s) used for a file in the file  
28119 system. Therefore, *pax* has not been given a mechanism to generate charset records  
28120 automatically. The only portable means of doing this is for the user to write the archive using the

28121       **-ocharset=string** command line option. This assumes that all of the files in the archive use the  
28122        same encoding. The “implementation-defined” text is included to allow for a system that can  
28123        identify the encodings used for each of its files.

28124       The table of standards that accompanies the charset record description is acknowledged to be  
28125        very limited. Only a limited number of character set standards is reasonable for maximal  
28126        interchange. Any character set is, of course, possible by prior agreement. It was suggested that  
28127        EBCDIC be listed, but it was omitted because it is not defined by a formal standard. Formal  
28128        standards, and then only those with reasonably large followings, can be included here, simply as  
28129        a matter of practicality. The <value>s represent names of officially registered character sets in the  
28130        format required by the ISO 2375:1985 standard.

28131       The normal comma or <blank>-separated list rules are not followed in the case of keyword  
28132        options to allow ease of argument parsing for *getopts*.

28133       Further information on character encodings is in **pax Archive Character Set Encoding/Decoding**  
28134       (on page 729).

28135       The standard developers have reserved keyword name space for vendor extensions. It is  
28136        suggested that the format to be used is:

28137       **VENDOR.keyword**

28138       where **VENDOR** is the name of the vendor or organization in all uppercase letters. It is further  
28139        suggested that the keyword following the period be named differently than any of the standard  
28140        keywords so that it could be used for future standardization, if appropriate, by omitting the  
28141        **VENDOR** prefix.

28142       The <length> field in the extended header record was included to make it simpler to step  
28143        through the records, even if a record contains an unknown format (to a particular *pax*) with  
28144        complex interactions of special characters. It also provides a minor integrity checkpoint within  
28145        the records to aid a program attempting to recover files from a damaged archive.

28146       There are no extended header versions of the *devmajor* and *devminor* fields because the  
28147        unspecified format **ustar** header field should be sufficient. If they are not, vendor-specific  
28148        extended keywords (such as **VENDOR.devmajor**) should be used.

28149       Device and *i*-number labeling of files was not adopted from *cpio*; files are interchanged strictly  
28150        on a symbolic name basis, as in **ustar**.

28151       Just as with the **ustar** format descriptions, the new format makes no special arrangements for  
28152        multi-volume archives. Each of the *pax* archive types is assumed to be inside a single POSIX file  
28153        and splitting that file over multiple volumes (diskettes, tape cartridges, and so on), processing  
28154        their labels, and mounting each in the proper sequence are considered to be implementation  
28155        details that cannot be described portably.

28156       The **pax** format is intended for interchange, not only for backup on a single (family of) systems.  
28157       It is not as densely packed as might be possible for backup:

- 28158       • It contains information as coded characters that could be coded in binary.
- 28159       • It identifies extended records with name fields that could be omitted in favor of a fixed-field  
28160        layout.
- 28161       • It translates names into a portable character set and identifies locale-related information,  
28162        both of which are probably unnecessary for backup.

28163       The requirements on restoring from an archive are slightly different from the historical wording,  
28164       allowing for non-monolithic privilege to bring forward as much as possible. In particular,  
28165       attributes such as “high performance file” might be broadly but not universally granted while

28166 set-user-ID or *chown()* might be much more restricted. There is no implication in  
28167 IEEE Std 1003.1-2001 that the security information be honored after it is restored to the file  
28168 hierarchy, in spite of what might be improperly inferred by the silence on that topic. That is a  
28169 topic for another standard.

28170 Links are recorded in the fashion described here because a link can be to any file type. It is  
28171 desirable in general to be able to restore part of an archive selectively and restore all of those  
28172 files completely. If the data is not associated with each link, it is not possible to do this.  
28173 However, the data associated with a file can be large, and when selective restoration is not  
28174 needed, this can be a significant burden. The archive is structured so that files that have no  
28175 associated data can always be restored by the name of any link name of any link, and the user  
28176 may choose whether data is recorded with each instance of a file that contains data. The format  
28177 permits mixing of both types of links in a single archive; this can be done for special needs, and  
28178 *pax* is expected to interpret such archives on input properly, despite the fact that there is no *pax*  
28179 option that would force this mixed case on output. (When **-o linkdata** is used, the output must  
28180 contain the duplicate data, but the implementation is free to include it or omit it when **-o**  
28181 **linkdata** is not used.)

28182 The time values are included as extended header records for those implementations needing  
28183 more than the eleven octal digits allowed by the **ustar** format. Portable file timestamps cannot be  
28184 negative. If *pax* encounters a file with a negative timestamp in **copy** or **write** mode, it can reject  
28185 the file, substitute a non-negative timestamp, or generate a non-portable timestamp with a  
28186 leading '-'. Even though some implementations can support finer file-time granularities than  
28187 seconds, the normative text requires support only for seconds since the Epoch because the  
28188 ISO POSIX-1 standard states them that way. The **ustar** format includes only *mtime*; the new  
28189 format adds *atime* and *ctime* for symmetry. The *atime* access time restored to the file system will  
28190 be affected by the **-p a** and **-p e** options. The *ctime* creation time (actually *inode* modification  
28191 time) is described with "appropriate privilege" so that it can be ignored when writing to the file  
28192 system. POSIX does not provide a portable means to change file creation time. Nothing is  
28193 intended to prevent a non-portable implementation of *pax* from restoring the value.

28194 The *gid*, *size*, and *uid* extended header records were included to allow expansion beyond the  
28195 sizes specified in the regular *tar* header. New file system architectures are emerging that will  
28196 exhaust the 12-digit size field. There are probably not many systems requiring more than 8 digits  
28197 for user and group IDs, but the extended header values were included for completeness,  
28198 allowing overrides for all of the decimal values in the *tar* header.

28199 The standard developers intended to describe the effective results of *pax* with regard to file  
28200 ownerships and permissions; implementations are not restricted in timing or sequencing the  
28201 restoration of such, provided the results are as specified.

28202 Much of the text describing the extended headers refers to use in "write or copy modes". The  
28203 **copy** mode references are due to the normative text: "The effect of the copy shall be as if the  
28204 copied files were written to an archive file and then subsequently extracted ...". There is  
28205 certainly no way to test whether *pax* is actually generating the extended headers in **copy** mode,  
28206 but the effects must be as if it had.

28207       **pax Archive Character Set Encoding/Decoding**

28208       There is a need to exchange archives of files between systems of different native codesets.

28209       Filenames, group names, and user names must be preserved to the fullest extent possible when

28210       an archive is read on the receiving platform. Translation of the contents of files is not within the

28211       scope of the *pax* utility.

28212       There will also be the need to represent characters that are not available on the receiving

28213       platform. These unsupported characters cannot be automatically folded to the local set of

28214       characters due to the chance of collisions. This could result in overwriting previous extracted

28215       files from the archive or pre-existing files on the system.

28216       For these reasons, the codeset used to represent characters within the extended header records of

28217       the *pax* archive must be sufficiently rich to handle all commonly used character sets. The fields

28218       requiring translation include, at a minimum, filenames, user names, group names, and link

28219       pathnames. Implementations may wish to have localized extended keywords that use non-

28220       portable characters.

28221       The standard developers considered the following options:

- The archive creator specifies the well-defined name of the source codeset. The receiver must then recognize the codeset name and perform the appropriate translations to the destination codeset.
- The archive creator includes within the archive the character mapping table for the source codeset used to encode extended header records. The receiver must then read the character mapping table and perform the appropriate translations to the destination codeset.
- The archive creator translates the extended header records in the source codeset into a canonical form. The receiver must then perform the appropriate translations to the destination codeset.

28231       The approach that incorporates the name of the source codeset poses the problem of codeset

28232       name registration, and makes the archive useless to *pax* archive decoders that do not recognize

28233       that codeset.

28234       Because parts of an archive may be corrupted, the standard developers felt that including the

28235       character map of the source codeset was too fragile. The loss of this one key component could

28236       result in making the entire archive useless. (The difference between this and the global extended

28237       header decision was that the latter has a workaround—duplicating extended header records on

28238       unreliable media—but this would be too burdensome for large character set maps.)

28239       Both of the above approaches also put an undue burden on the *pax* archive receiver to handle the

28240       cross-product of all source and destination codesets.

28241       To simplify the translation from the source codeset to the canonical form and from the canonical

28242       form to the destination codeset, the standard developers decided that the internal representation

28243       should be a stateless encoding. A stateless encoding is one where each codepoint has the same

28244       meaning, without regard to the decoder being in a specific state. An example of a stateful

28245       encoding would be the Japanese Shift-JIS; an example of a stateless encoding would be the

28246       ISO/IEC 646: 1991 standard (equivalent to 7-bit ASCII).

28247       For these reasons, the standard developers decided to adopt a canonical format for the

28248       representation of file information strings. The obvious, well-endorsed candidate is the

28249       ISO/IEC 10646-1: 2000 standard (based in part on Unicode), which can be used to represent the

28250       characters of virtually all standardized character sets. The standard developers initially agreed

28251       upon using UCS2 (16-bit Unicode) as the internal representation. This repertoire of characters

28252       provides a sufficiently rich set to represent all commonly-used codesets.

28253 However, the standard developers found that the 16-bit Unicode representation had some  
28254 problems. It forced the issue of standardizing byte ordering. The 2-byte length of each character  
28255 made the extended header records twice as long for the case of strings coded entirely from  
28256 historical 7-bit ASCII. For these reasons, the standard developers chose the UTF-8 defined in the  
28257 ISO/IEC 10646-1: 2000 standard. This multi-byte representation encodes UCS2 or UCS4  
28258 characters reliably and deterministically, eliminating the need for a canonical byte ordering. In  
28259 addition, NUL octets and other characters possibly confusing to POSIX file systems do not  
28260 appear, except to represent themselves. It was realized that certain national codesets take up  
28261 more space after the encoding, due to their placement within the UCS range; it was felt that the  
28262 usefulness of the encoding of the names outweighs the disadvantage of size increase for file,  
28263 user, and group names.

28264 The encoding of UTF-8 is as follows:

28265   UCS4 Hex Encoding   UTF-8 Binary Encoding  
28266   00000000-0000007F   0xxxxxxx  
28267   00000080-0000007FF   110xxxxx 10xxxxxx  
28268   000000800-0000FFFF   1110xxxx 10xxxxxx 10xxxxxx  
28269   00010000-001FFFFF   11110xxx 10xxxxxx 10xxxxxx 10xxxxxx  
28270   00200000-03FFFFFF   111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx  
28271   04000000-7FFFFFFF   1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

28272 where each 'x' represents a bit value from the character being translated.

### 28273 **ustar Interchange Format**

28274 The description of the **ustar** format reflects numerous enhancements over pre-1988 versions of  
28275 the historical **tar** utility. The goal of these changes was not only to provide the functional  
28276 enhancements desired, but also to retain compatibility between new and old versions. This  
28277 compatibility has been retained. Archives written using the old archive format are compatible  
28278 with the new format.

28279 Implementors should be aware that the previous file format did not include a mechanism to  
28280 archive directory type files. For this reason, the convention of using a filename ending with slash  
28281 was adopted to specify a directory on the archive.

28282 The total size of the *name* and *prefix* fields have been set to meet the minimum requirements for  
28283 {PATH\_MAX}. If a pathname will fit within the *name* field, it is recommended that the pathname  
28284 be stored there without the use of the *prefix* field. Although the *name* field is known to be too  
28285 small to contain {PATH\_MAX} characters, the value was not changed in this version of the  
28286 archive file format to retain backwards-compatibility, and instead the *prefix* was introduced.  
28287 Also, because of the earlier version of the format, there is no way to remove the restriction on the  
28288 *linkname* field being limited in size to just that of the *name* field.

28289 The *size* field is required to be meaningful in all implementation extensions, although it could be  
28290 zero. This is required so that the data blocks can always be properly counted.

28291 It is suggested that if device special files need to be represented that cannot be represented in the  
28292 standard format, that one of the extension types (A-Z) be used, and that the additional  
28293 information for the special file be represented as data and be reflected in the *size* field.

28294 Attempting to restore a special file type, where it is converted to ordinary data and conflicts  
28295 with an existing filename, need not be specially detected by the utility. If run as an ordinary user,  
28296 *pax* should not be able to overwrite the entries in, for example, /**dev** in any case (whether the file  
28297 is converted to another type or not). If run as a privileged user, it should be able to do so, and it  
28298 would be considered a bug if it did not. The same is true of ordinary data files and similarly

28299 named special files; it is impossible to anticipate the needs of the user (who could really intend  
28300 to overwrite the file), so the behavior should be predictable (and thus regular) and rely on the  
28301 protection system as required.

28302 The value 7 in the *typeflag* field is intended to define how contiguous files can be stored in a  
28303 **ustar** archive. IEEE Std 1003.1-2001 does not require the contiguous file extension, but does  
28304 define a standard way of archiving such files so that all conforming systems can interpret these  
28305 file types in a meaningful and consistent manner. On a system that does not support extended  
28306 file types, the **pax** utility should do the best it can with the file and go on to the next.

28307 The file protection modes are those conventionally used by the *ls* utility. This is extended  
28308 beyond the usage in the ISO POSIX-2 standard to support the “shared text” or “sticky” bit. It is  
28309 intended that the conformance document should not document anything beyond the existence  
28310 of and support of such a mode. Further extensions are expected to these bits, particularly with  
28311 overloading the set-user-ID and set-group-ID flags.

### 28312 **cpio Interchange Format**

28313 The reference to appropriate privilege in the **cpio** format refers to an error on standard output;  
28314 the **ustar** format does not make comparable statements.

28315 The model for this format was the historical System V *cpio-c* data interchange format. This  
28316 model documents the portable version of the **cpio** format and not the binary version. It has the  
28317 flexibility to transfer data of any type described within IEEE Std 1003.1-2001, yet is extensible to  
28318 transfer data types specific to extensions beyond IEEE Std 1003.1-2001 (for example, contiguous  
28319 files). Because it describes existing practice, there is no question of maintaining upwards-  
28320 compatibility.

### 28321 **cpio Header**

28322 There has been some concern that the size of the *c\_ino* field of the header is too small to handle  
28323 those systems that have very large *inode* numbers. However, the *c\_ino* field in the header is used  
28324 strictly as a hard-link resolution mechanism for archives. It is not necessarily the same value as  
28325 the *inode* number of the file in the location from which that file is extracted.

28326 The name *c\_magic* is based on historical usage.

### 28327 **cpio Filename**

28328 For most historical implementations of the **cpio** utility, {PATH\_MAX} octets can be used to  
28329 describe the pathname without the addition of any other header fields (the NUL character  
28330 would be included in this count). {PATH\_MAX} is the minimum value for pathname size,  
28331 documented as 256 bytes. However, an implementation may use *c\_namesize* to determine the  
28332 exact length of the pathname. With the current description of the <**cpio.h**> header, this  
28333 pathname size can be as large as a number that is described in six octal digits.

28334 Two values are documented under the *c\_mode* field values to provide for extensibility for known  
28335 file types:

28336 **0110 000** Reserved for contiguous files. The implementation may treat the rest of the  
28337 information for this archive like a regular file. If this file type is undefined, the  
28338 implementation may create the file as a regular file.

28339 This provides for extensibility of the **cpio** format while allowing for the ability to read old  
28340 archives. Files of an unknown type may be read as “regular files” on some implementations. On  
28341 a system that does not support extended file types, the **pax** utility should do the best it can with  
28342 the file and go on to the next.

28343 **FUTURE DIRECTIONS**

28344 None.

28345 **SEE ALSO**28346 Chapter 2 (on page 29), *cp*, *ed*, *getopts*, *ls*, *printf*, the Base Definitions volume of  
28347 IEEE Std 1003.1-2001, <**cpio.h**>, the System Interfaces volume of IEEE Std 1003.1-2001, *chown()*,  
28348 *creat()*, *mkdir()*, *mkfifo()*, *stat()*, *utime()*, *write()*28349 **CHANGE HISTORY**

28350 First released in Issue 4.

28351 **Issue 5**28352 A note is added to the APPLICATION USAGE indicating that the **cpio** and **tar** formats can only  
28353 support files up to 8 gigabytes in size.28354 **Issue 6**28355 The *pax* utility is aligned with the IEEE P1003.2b draft standard:

- 28356 • Support has been added for symbolic links in the options and interchange formats.
- 28357 • A new format has been devised, based on extensions to **ustar**.
- 28358 • References to the “extended” **tar** and **cpio** formats derived from the POSIX.1-1990 standard  
28359 have been changed to remove the “extended” adjective because this could cause confusion  
28360 with the extended **tar** header added in this revision. (All references to **tar** are actually to  
28361 **ustar**.)

28362 The **TZ** entry is added to the ENVIRONMENT VARIABLES section.28363 IEEE PASC Interpretation 1003.2 #168 is applied, clarifying that *mkdir()* and *mkfifo()* calls can  
28364 ignore an [EEXIST] error when extracting an archive.28365 IEEE PASC Interpretation 1003.2 #180 is applied, clarifying how extracted files are created when  
28366 in **read** mode.28367 IEEE PASC Interpretation 1003.2 #181 is applied, clarifying the description of the **-t** option.

28368 IEEE PASC Interpretation 1003.2 #195 is applied.

28369 IEEE PASC Interpretation 1003.2 #206 is applied, clarifying the handling of links for the **-H**, **-L**,  
28370 and **-I** options.

## 28371 NAME

28372 pr — print files

## 28373 SYNOPSIS

```
28374 pr [+page][-column][-adFmrt][-e[char][gap]][-h header][-i[char][gap]]
28375 XSI [-l lines][-n[char][width]][-o offset][-s[char]][-w width][-fp]
28376 [file...]
```

## 28377 DESCRIPTION

28378 The *pr* utility is a printing and pagination filter. If multiple input files are specified, each shall be  
 28379 read, formatted, and written to standard output. By default, the input shall be separated into 66-line  
 28380 pages, each with:

- 28381 • A 5-line header that includes the page number, date, time, and the pathname of the file
- 28382 • A 5-line trailer consisting of blank lines

28383 If standard output is associated with a terminal, diagnostic messages shall be deferred until the  
 28384 *pr* utility has completed processing.

28385 When options specifying multi-column output are specified, output text columns shall be of  
 28386 equal width; input lines that do not fit into a text column shall be truncated. By default, text  
 28387 columns shall be separated with at least one <blank>.

## 28388 OPTIONS

28389 The *pr* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
 28390 Utility Syntax Guidelines, except that: the *page* option has a '+' delimiter; *page* and *column* can  
 28391 be multi-digit numbers; some of the option-arguments are optional; and some of the option-  
 28392 arguments cannot be specified as separate arguments from the preceding option letter. In  
 28393 particular, the **-s** option does not allow the option letter to be separated from its argument, and  
 28394 the options **-e**, **-i**, and **-n** require that both arguments, if present, not be separated from the  
 28395 option letter.

28396 The following options shall be supported. In the following option descriptions, *column*, *lines*,  
 28397 *offset*, *page*, and *width* are positive decimal integers; *gap* is a non-negative decimal integer.

- 28398 **+page** Begin output at page number *page* of the formatted input.
- 28399 **-column** Produce multi-column output that is arranged in *column* columns (the default shall  
 28400 be 1) and is written down each column in the order in which the text is received  
 28401 from the input file. This option should not be used with **-m**. The options **-e** and **-i**  
 28402 shall be assumed for multiple text-column output. Whether or not text columns  
 28403 are produced with identical vertical lengths is unspecified, but a text column shall  
 28404 never exceed the length of the page (see the **-l** option). When used with **-t**, use the  
 28405 minimum number of lines to write the output.
- 28406 **-a** Modify the effect of the **-column** option so that the columns are filled across the  
 28407 page in a round-robin order (for example, when *column* is 2, the first input line  
 28408 heads column 1, the second heads column 2, the third is the second line in column  
 28409 1, and so on).
- 28410 **-d** Produce output that is double-spaced; append an extra <newline> following every  
 28411 <newline> found in the input.
- 28412 **-e[char][gap]** Expand each input <tab> to the next greater column position specified by the  
 28413 formula  $n * gap + 1$ , where *n* is an integer  $> 0$ . If *gap* is zero or is omitted, it shall  
 28414 default to 8. All <tab>s in the input shall be expanded into the appropriate number  
 28415 of <space>s. If any non-digit character, *char*, is specified, it shall be used as the

|           |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28417     |                        | input <tab>.                                                                                                                                                                                                                                                                                                                                                                                    |
| 28418 XSI | <b>-f</b>              | Use a <form-feed> for new pages, instead of the default behavior that uses a sequence of <newline>s. Pause before beginning the first page if the standard output is associated with a terminal.                                                                                                                                                                                                |
| 28419     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28420     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28421     | <b>-F</b>              | Use a <form-feed> for new pages, instead of the default behavior that uses a sequence of <newline>s.                                                                                                                                                                                                                                                                                            |
| 28422     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28423     | <b>-h header</b>       | Use the string <i>header</i> to replace the contents of the <i>file</i> operand in the page header.                                                                                                                                                                                                                                                                                             |
| 28424     | <b>-i[char][gap]</b>   | In output, replace multiple <space>s with <tab>s wherever two or more adjacent <space>s reach column positions <i>gap</i> +1, 2* <i>gap</i> +1, 3* <i>gap</i> +1, and so on. If <i>gap</i> is zero or is omitted, default tab settings at every eighth column position shall be assumed. If any non-digit character, <i>char</i> , is specified, it shall be used as the output <tab>.          |
| 28425     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28426     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28427     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28428     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28429     | <b>-l lines</b>        | Override the 66-line default and reset the page length to <i>lines</i> . If <i>lines</i> is not greater than the sum of both the header and trailer depths (in lines), the <i>pr</i> utility shall suppress both the header and trailer, as if the <b>-t</b> option were in effect.                                                                                                             |
| 28430     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28431     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28432     | <b>-m</b>              | Merge files. Standard output shall be formatted so the <i>pr</i> utility writes one line from each file specified by a <i>file</i> operand, side by side into text columns of equal fixed widths, in terms of the number of column positions. Implementations shall support merging of at least nine <i>file</i> operands.                                                                      |
| 28433     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28434     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28435     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28436     | <b>-n[char][width]</b> | Provide <i>width</i> -digit line numbering (default for <i>width</i> shall be 5). The number shall occupy the first <i>width</i> column positions of each text column of default output or each line of <b>-m</b> output. If <i>char</i> (any non-digit character) is given, it shall be appended to the line number to separate it from whatever follows (default for <i>char</i> is a <tab>). |
| 28437     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28438     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28439     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28440     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28441     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28442     | <b>-o offset</b>       | Each line of output shall be preceded by offset <space>s. If the <b>-o</b> option is not specified, the default offset shall be zero. The space taken is in addition to the output line width (see the <b>-w</b> option below).                                                                                                                                                                 |
| 28443     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28444     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28445     | <b>-p</b>              | Pause before beginning each page if the standard output is directed to a terminal ( <i>pr</i> shall write an <alert> to standard error and wait for a <carriage-return> to be read on /dev/tty).                                                                                                                                                                                                |
| 28446     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28447     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28448     | <b>-r</b>              | Write no diagnostic reports on failure to open files.                                                                                                                                                                                                                                                                                                                                           |
| 28449     | <b>-s[char]</b>        | Separate text columns by the single character <i>char</i> instead of by the appropriate number of <space>s (default for <i>char</i> shall be <tab>).                                                                                                                                                                                                                                            |
| 28450     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28451     | <b>-t</b>              | Write neither the five-line identifying header nor the five-line trailer usually supplied for each page. Quit writing after the last line of each file without spacing to the end of the page.                                                                                                                                                                                                  |
| 28452     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28453     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28454     | <b>-w width</b>        | Set the width of the line to <i>width</i> column positions for multiple text-column output only. If the <b>-w</b> option is not specified and the <b>-s</b> option is not specified, the default width shall be 72. If the <b>-w</b> option is not specified and the <b>-s</b> option is specified, the default width shall be 512.                                                             |
| 28455     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28456     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28457     |                        |                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28458     |                        | For single column output, input lines shall not be truncated.                                                                                                                                                                                                                                                                                                                                   |

28459 **OPERANDS**

28460 The following operand shall be supported:

28461 *file* A pathname of a file to be written. If no *file* operands are specified, or if a *file* operand is ‘–’, the standard input shall be used.28463 **STDIN**28464 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is ‘–’.  
28465 See the INPUT FILES section.28466 **INPUT FILES**

28467 The input files shall be text files.

28468 The file /dev/tty shall be used to read responses required by the –p option.

28469 **ENVIRONMENT VARIABLES**28470 The following environment variables shall affect the execution of *pr*:28471 *LANG* Provide a default value for the internationalization variables that are unset or null.  
28472 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
28473 Internationalization Variables for the precedence of internationalization variables  
28474 used to determine the values of locale categories.)28475 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
28476 internationalization variables.28477 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
28478 characters (for example, single-byte as opposed to multi-byte characters in  
28479 arguments and input files) and which characters are defined as printable (character  
28480 class **print**). Non-printable characters are still written to standard output, but are  
28481 not counted for the purpose for column-width and line-length calculations.28482 *LC\_MESSAGES*28483 Determine the locale that should be used to affect the format and contents of  
28484 diagnostic messages written to standard error.28485 *LC\_TIME* Determine the format of the date and time for use in writing header lines.28486 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.28487 *TZ* Determine the timezone used to calculate date and time strings written in header  
28488 lines. If *TZ* is unset or null, an unspecified default timezone shall be used.28489 **ASYNCHRONOUS EVENTS**28490 If *pr* receives an interrupt while writing to a terminal, it shall flush all accumulated error  
28491 messages to the screen before terminating.28492 **STDOUT**28493 The *pr* utility output shall be a paginated version of the original file (or files). This pagination  
28494 shall be accomplished using either <form-feed>s or a sequence of <newline>s, as controlled by  
28495 the –F or –f option. Page headers shall be generated unless the –t option is specified. The page  
28496 headers shall be of the form:28497 "*\n\n%s %s Page %d\n\n\n*", <output of date>, <file>, <page number>28498 In the POSIX locale, the <output of date> field, representing the date and time of last modification  
28499 of the input file (or the current date and time if the input file is standard input), shall be  
28500 equivalent to the output of the following command as it would appear if executed at the given  
28501 time:

28502 date "+%b %e %H:%M %Y"  
28503 without the trailing <newline>, if the page being written is from standard input. If the page  
28504 being written is not from standard input, in the POSIX locale, the same format shall be used, but  
28505 the time used shall be the modification time of the file corresponding to *file* instead of the current  
28506 time. When the *LC\_TIME* locale category is not set to the POSIX locale, a different format and  
28507 order of presentation of this field may be used.

28508 If the standard input is used instead of a *file* operand, the <*file*> field shall be replaced by a null  
28509 string.

28510 If the **-h** option is specified, the <*file*> field shall be replaced by the *header* argument.

## 28511 **STDERR**

28512 The standard error shall be used for diagnostic messages and for alerting the terminal when **-p**  
28513 is specified.

## 28514 **OUTPUT FILES**

28515 None.

## 28516 **EXTENDED DESCRIPTION**

28517 None.

## 28518 **EXIT STATUS**

28519 The following exit values shall be returned:

28520 0 Successful completion.

28521 >0 An error occurred.

## 28522 **CONSEQUENCES OF ERRORS**

28523 Default.

## 28524 **APPLICATION USAGE**

28525 None.

## 28526 **EXAMPLES**

28527 1. Print a numbered list of all files in the current directory:

28528 ls -a | pr -n -h "Files in \$(pwd)."

28529 2. Print **file1** and **file2** as a double-spaced, three-column listing headed by "file list":

28530 pr -3d -h "file list" file1 file2

28531 3. Write **file1** on **file2**, expanding tabs to columns 10, 19, 28, ...:

28532 pr -e9 -t <file1 >file2

## 28533 **RATIONALE**

28534 This utility is one of those that does not follow the Utility Syntax Guidelines because of its  
28535 historical origins. The standard developers could have added new options that obeyed the  
28536 guidelines (and marked the old options obsolescent) or devised an entirely new utility; there are  
28537 examples of both actions in this volume of IEEE Std 1003.1-2001. Because of its widespread use  
28538 by historical applications, the standard developers decided to exempt this version of *pr* from  
28539 many of the guidelines.

28540 Implementations are required to accept option-arguments to the **-h**, **-l**, **-o**, and **-w** options  
28541 whether presented as part of the same argument or as a separate argument to *pr*, as suggested by  
28542 the Utility Syntax Guidelines. The **-n** and **-s** options, however, are specified as in historical  
28543 practice because they are frequently specified without their optional arguments. If a <blank>

28544 were allowed before the option-argument in these cases, a *file* operand could mistakenly be  
28545 interpreted as an option-argument in historical applications.

28546 The text about the minimum number of lines in multi-column output was included to ensure  
28547 that a best effort is made in balancing the length of the columns. There are known historical  
28548 implementations in which, for example, 60-line files are listed by *pr -2* as one column of 56 lines  
28549 and a second of 4. Although this is not a problem when a full page with headers and trailers is  
28550 produced, it would be relatively useless when used with *-t*.

28551 Historical implementations of the *pr* utility have differed in the action taken for the *-f* option.  
28552 BSD uses it as described here for the *-F* option; System V uses it to change trailing <newline>s  
28553 on each page to a <form-feed> and, if standard output is a TTY device, sends an <alert> to  
28554 standard error and reads a line from /dev/tty before the first page. There were strong arguments  
28555 from both sides of this issue concerning historical practice and as a result the *-F* option was  
28556 added. XSI-conformant systems support the System V historical actions for the *-f* option.

28557 The <output of date> field in the *-l* format is specified only for the POSIX locale. As noted, the  
28558 format can be different in other locales. No mechanism for defining this is present in this volume  
28559 of IEEE Std 1003.1-2001, as the appropriate vehicle is a message catalog; that is, the format  
28560 should be specified as a “message”.

## 28561 FUTURE DIRECTIONS

28562 None.

## 28563 SEE ALSO

28564 *expand, lp*

## 28565 CHANGE HISTORY

28566 First released in Issue 2.

## 28567 Issue 6

28568 The following new requirements on POSIX implementations derive from alignment with the  
28569 Single UNIX Specification:

- 28570 • The *-p* option is added.

28571 The normative text is reworded to avoid use of the term “must” for application requirements.

**28572 NAME**

28573        printf — write formatted output

**28574 SYNOPSIS**

28575        `printf format[argument...]`

**28576 DESCRIPTION**

28577        The *printf* utility shall write formatted operands to the standard output. The *argument* operands  
28578        shall be formatted under control of the *format* operand.

**28579 OPTIONS**

28580        None.

**28581 OPERANDS**

28582        The following operands shall be supported:

28583        *format*        A string describing the format to use to write the remaining operands. See the  
28584                    EXTENDED DESCRIPTION section.

28585        *argument*     The strings to be written to standard output, under the control of *format*. See the  
28586                    EXTENDED DESCRIPTION section.

**28587 STDIN**

28588        Not used.

**28589 INPUT FILES**

28590        None.

**28591 ENVIRONMENT VARIABLES**

28592        The following environment variables shall affect the execution of *printf*.

28593        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
28594                    (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
28595                    Internationalization Variables for the precedence of internationalization variables  
28596                    used to determine the values of locale categories.)

28597        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
28598                    internationalization variables.

28599        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
28600                    characters (for example, single-byte as opposed to multi-byte characters in  
28601                    arguments).

**28602 *LC\_MESSAGES***

28603        Determine the locale that should be used to affect the format and contents of  
28604                    diagnostic messages written to standard error.

**28605 *LC\_NUMERIC***

28606        Determine the locale for numeric formatting. It shall affect the format of numbers  
28607                    written using the e, E, f, g, and G conversion specifier characters (if supported).

28608 XSI      *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**28609 ASYNCHRONOUS EVENTS**

28610        Default.

**28611 STDOUT**

28612        See the EXTENDED DESCRIPTION section.

28613 **STDERR**

28614 The standard error shall be used only for diagnostic messages.

28615 **OUTPUT FILES**

28616 None.

28617 **EXTENDED DESCRIPTION**28618 The *format* operand shall be used as the *format* string described in the Base Definitions volume of  
28619 IEEE Std 1003.1-2001, Chapter 5, File Format Notation with the following exceptions:

- 28620 1. A <space> in the format string, in any context other than a flag of a conversion  
28621 specification, shall be treated as an ordinary character that is copied to the output.
- 28622 2. A 'Δ' character in the format string shall be treated as a 'Δ' character, not as a <space>.
- 28623 3. In addition to the escape sequences shown in the Base Definitions volume of  
28624 IEEE Std 1003.1-2001, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n',  
28625 '\r', '\t', '\v'), "\ddd", where *ddd* is a one, two, or three-digit octal number, shall be  
28626 written as a byte with the numeric value specified by the octal number.
- 28627 4. The implementation shall not precede or follow output from the d or u conversion  
28628 specifiers with <blank>s not specified by the *format* operand.
- 28629 5. The implementation shall not precede output from the o conversion specifier with zeros  
28630 not specified by the *format* operand.
- 28631 6. The e, E, f, g, and G conversion specifiers need not be supported.
- 28632 7. An additional conversion specifier character, b, shall be supported as follows. The  
28633 argument shall be taken to be a string that may contain backslash-escape sequences. The  
28634 following backslash-escape sequences shall be supported:
  - 28635 — The escape sequences listed in the Base Definitions volume of IEEE Std 1003.1-2001,  
28636 Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v'),  
28637 which shall be converted to the characters they represent
  - 28638 — "\0ddd", where *ddd* is a zero, one, two, or three-digit octal number that shall be  
28639 converted to a byte with the numeric value specified by the octal number
  - 28640 — '\c', which shall not be written and shall cause *printf* to ignore any remaining  
28641 characters in the string operand containing it, any remaining string operands, and any  
28642 additional characters in the *format* operand

28643 The interpretation of a backslash followed by any other sequence of characters is  
28644 unspecified.28645 Bytes from the converted string shall be written until the end of the string or the number of  
28646 bytes indicated by the precision specification is reached. If the precision is omitted, it shall  
28647 be taken to be infinite, so all bytes up to the end of the converted string shall be written.

- 28648 8. For each conversion specification that consumes an argument, the next argument operand  
28649 shall be evaluated and converted to the appropriate type for the conversion as specified  
28650 below.
- 28651 9. The *format* operand shall be reused as often as necessary to satisfy the argument operands.  
28652 Any extra c or s conversion specifiers shall be evaluated as if a null string argument were  
28653 supplied; other extra conversion specifications shall be evaluated as if a zero argument  
28654 were supplied. If the *format* operand contains no conversion specifications and *argument*  
28655 operands are present, the results are unspecified.

28656        10. If a character sequence in the *format* operand begins with a '%' character, but does not  
28657        form a valid conversion specification, the behavior is unspecified.

28658        The *argument* operands shall be treated as strings if the corresponding conversion specifier is b,  
28659        c, or s; otherwise, it shall be evaluated as a C constant, as described by the ISO C standard, with  
28660        the following extensions:

- 28661        • A leading plus or minus sign shall be allowed.
- 28662        • If the leading character is a single-quote or double-quote, the value shall be the numeric  
28663        value in the underlying codeset of the character following the single-quote or double-quote.

28664        If an argument operand cannot be completely converted into an internal value appropriate to  
28665        the corresponding conversion specification, a diagnostic message shall be written to standard  
28666        error and the utility shall not exit with a zero exit status, but shall continue processing any  
28667        remaining operands and shall write the value accumulated at the time the error was detected to  
28668        standard output.

28669        It is not considered an error if an argument operand is not completely used for a c or s  
28670        conversion or if a string operand's first or second character is used to get the numeric value of a  
28671        character.

#### 28672 EXIT STATUS

28673        The following exit values shall be returned:

- 28674        0 Successful completion.
- 28675        >0 An error occurred.

#### 28676 CONSEQUENCES OF ERRORS

28677        Default.

#### 28678 APPLICATION USAGE

28679        The floating-point formatting conversion specifications of *printf()* are not required because all  
28680        arithmetic in the shell is integer arithmetic. The awk utility performs floating-point calculations  
28681        and provides its own printf function. The bc utility can perform arbitrary-precision floating-  
28682        point arithmetic, but does not provide extensive formatting capabilities. (This printf utility  
28683        cannot really be used to format bc output; it does not support arbitrary precision.)  
28684        Implementations are encouraged to support the floating-point conversions as an extension.

28685        Note that this printf utility, like the printf() function defined in the System Interfaces volume of  
28686        IEEE Std 1003.1-2001 on which it is based, makes no special provision for dealing with multi-  
28687        byte characters when using the %c conversion specification or when a precision is specified in a  
28688        %b or %s conversion specification. Applications should be extremely cautious using either of  
28689        these features when there are multi-byte characters in the character set.

28690        No provision is made in this volume of IEEE Std 1003.1-2001 which allows field widths and  
28691        precisions to be specified as '\*' since the '\*' can be replaced directly in the *format* operand  
28692        using shell variable substitution. Implementations can also provide this feature as an extension  
28693        if they so choose.

28694        Hexadecimal character constants as defined in the ISO C standard are not recognized in the  
28695        *format* operand because there is no consistent way to detect the end of the constant. Octal  
28696        character constants are limited to, at most, three octal digits, but hexadecimal character  
28697        constants are only terminated by a non-hex-digit character. In the ISO C standard, the "##"  
28698        concatenation operator can be used to terminate a constant and follow it with a hexadecimal  
28699        character to be written. In the shell, concatenation occurs before the printf utility has a chance to  
28700        parse the end of the hexadecimal constant.

28701 The %b conversion specification is not part of the ISO C standard; it has been added here as a  
 28702 portable way to process backslash escapes expanded in string operands as provided by the echo  
 28703 utility. See also the APPLICATION USAGE section of echo (on page 331) for ways to use printf as  
 28704 a replacement for all of the traditional versions of the echo utility.

28705 If an argument cannot be parsed correctly for the corresponding conversion specification, the  
 28706 printf utility is required to report an error. Thus, overflow and extraneous characters at the end  
 28707 of an argument being used for a numeric conversion shall be reported as errors.

## 28708 EXAMPLES

28709 To alert the user and then print and read a series of prompts:

```
28710 printf "\aPlease fill in the following: \nName: "
28711 read name
28712 printf "Phone number: "
28713 read phone
```

28714 To read out a list of right and wrong answers from a file, calculate the percentage correctly, and  
 28715 print them out. The numbers are right-justified and separated by a single <tab>. The percentage  
 28716 is written to one decimal place of accuracy:

```
28717 while read right wrong ; do
28718 percent=$(echo "scale=1;($right*100)/($right+$wrong)" | bc)
28719 printf "%2d right\t%2d wrong\t(%s%%)\n" \
28720 $right $wrong $percent
28721 done < database_file
```

28722 The command:

```
28723 printf "%5d%4d\n" 1 21 321 4321 54321
```

28724 produces:

```
28725 1 21
28726 3214321
28727 54321 0
```

28728 Note that the *format* operand is used three times to print all of the given strings and that a '0'  
 28729 was supplied by printf to satisfy the last %4d conversion specification.

28730 The printf utility is required to notify the user when conversion errors are detected while  
 28731 producing numeric output; thus, the following results would be expected on an implementation  
 28732 with 32-bit two's-complement integers when %d is specified as the *format* operand:

| 28733 Argument    | 28734 Standard Output | 28735 Diagnostic Output                        |
|-------------------|-----------------------|------------------------------------------------|
| 28735 5a          | 28736 5               | 28737 printf: "5a" not completely converted    |
| 28736 9999999999  | 28737 2147483647      | 28738 printf: "9999999999" arithmetic overflow |
| 28737 -9999999999 | 28738 -2147483648     | printf: "-9999999999" arithmetic overflow      |
| 28738 ABC         | 0                     | printf: "ABC" expected numeric value           |

28739 The diagnostic message format is not specified, but these examples convey the type of  
 28740 information that should be reported. Note that the value shown on standard output is what  
 28741 would be expected as the return value from the strtol() function as defined in the System  
 28742 Interfaces volume of IEEE Std 1003.1-2001. A similar correspondence exists between %u and  
 28743 strtoul() and %e, %f, and %g (if the implementation supports floating-point conversions) and  
 28744 strtod().

28745 In a locale using the ISO/IEC 646: 1991 standard as the underlying codeset, the command:  
28746 `printf "%d\n" 3 +3 -3 \'3 \'+"3 "'-3"`  
28747 produces:  
28748   3   Numeric value of constant 3  
28749   3   Numeric value of constant 3  
28750   -3   Numeric value of constant -3  
28751   51   Numeric value of the character '3' in the ISO/IEC 646: 1991 standard codeset  
28752   43   Numeric value of the character '+' in the ISO/IEC 646: 1991 standard codeset  
28753   45   Numeric value of the character '-' in the ISO/IEC 646: 1991 standard codeset  
28754 Note that in a locale with multi-byte characters, the value of a character is intended to be the  
28755 value of the equivalent of the `wchar_t` representation of the character as described in the System  
28756 Interfaces volume of IEEE Std 1003.1-2001.

#### 28757 RATIONALE

28758 The *printf* utility was added to provide functionality that has historically been provided by *echo*.  
28759 However, due to irreconcilable differences in the various versions of *echo* extant, the version has  
28760 few special features, leaving those to this new *printf* utility, which is based on one in the Ninth  
28761 Edition system.

28762 The EXTENDED DESCRIPTION section almost exactly matches the *printf()* function in the  
28763 ISO C standard, although it is described in terms of the file format notation in the Base  
28764 Definitions volume of IEEE Std 1003.1-2001, Chapter 5, File Format Notation.

#### 28765 FUTURE DIRECTIONS

28766 None.

#### 28767 SEE ALSO

28768 *awk*, *bc*, *echo*, the System Interfaces volume of IEEE Std 1003.1-2001, *printf()*

#### 28769 CHANGE HISTORY

28770 First released in Issue 4.

## 28771 NAME

28772 prs — print an SCCS file (**DEVELOPMENT**)

## 28773 SYNOPSIS

28774 XSI prs [-a][-d dataspec][-r[SID]] file...

28775 XSI prs [-e| -l] -c cutoff [-d dataspec] file...

28776 XSI prs [-e| -l] -r[SID][-d dataspec]file...

## 28777 DESCRIPTION

28778 The *prs* utility shall write to standard output parts or all of an SCCS file in a user-supplied  
28779 format.

## 28780 OPTIONS

28781 The *prs* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
28782 Utility Syntax Guidelines, except that the **-r** option has an optional option-argument. This  
28783 optional option-argument cannot be presented as a separate argument. The following options  
28784 shall be supported:28785 **-d** *dataspec* Specify the output data specification. The *dataspec* shall be a string consisting of  
28786 SCCS file *data keywords* (see **Data Keywords** (on page 744)) interspersed with  
28787 optional user-supplied text.28788 **-r**[*SID*] Specify the SCCS identification string (SID) of a delta for which information is  
28789 desired. If no *SID* option-argument is specified, the SID of the most recently  
2890 created delta shall be assumed.28791 **-e** Request information for all deltas created earlier than and including the delta  
28792 designated via the **-r** option or the date-time given by the **-c** option.28793 **-l** Request information for all deltas created later than and including the delta  
28794 designated via the **-r** option or the date-time given by the **-c** option.28795 **-c** *cutoff* Indicate the *cutoff* date-time, in the form:

28796 YY[MM[DD[HH[MM[SS]]]]]

28797 For the YY component, values in the range [69,99] shall refer to years 1969 to 1999  
28798 inclusive, and values in the range [00,68] shall refer to years 2000 to 2068 inclusive.28799 **Note:** It is expected that in a future version of IEEE Std 1003.1-2001 the default  
2880 century inferred from a 2-digit year will change. (This would apply to all  
2881 commands accepting a 2-digit year as input.)28802 No changes (deltas) to the SCCS file that were created after the specified *cutoff*  
28803 date-time shall be included in the output. Units omitted from the date-time default  
28804 to their maximum possible values; for example, **-c 7502** is equivalent to  
28805 **-c 750228235959**.28806 **-a** Request writing of information for both removed—that is, *delta type=R* (see  
28807 *rmdel*)—and existing—that is, *delta type=D*,—deltas. If the **-a** option is not  
28808 specified, information for existing deltas only shall be provided.

## 28809 OPERANDS

28810 The following operand shall be supported:

28811 *file* A pathname of an existing SCCS file or a directory. If *file* is a directory, the *prs*  
28812 utility shall behave as though each file in the directory were specified as a named  
28813 file, except that non-SCCS files (last component of the pathname does not begin  
28814 with **s.**) and unreadable files shall be silently ignored.

28815 If exactly one *file* operand appears, and it is ‘–’, the standard input shall be read;  
28816 each line of the standard input shall be taken to be the name of an SCCS file to be  
28817 processed. Non-SCCS files and unreadable files shall be silently ignored.

## 28818 **STDIN**

28819 The standard input shall be a text file used only when the *file* operand is specified as ‘–’. Each  
28820 line of the text file shall be interpreted as an SCCS pathname.

## 28821 **INPUT FILES**

28822 Any SCCS files displayed are files of an unspecified format.

## 28823 **ENVIRONMENT VARIABLES**

28824 The following environment variables shall affect the execution of *prs*:

28825 *LANG* Provide a default value for the internationalization variables that are unset or null.  
28826 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
28827 Internationalization Variables for the precedence of internationalization variables  
28828 used to determine the values of locale categories.)

28829 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
28830 internationalization variables.

28831 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
28832 characters (for example, single-byte as opposed to multi-byte characters in  
28833 arguments and input files).

28834 *LC\_MESSAGES*

28835 Determine the locale that should be used to affect the format and contents of  
28836 diagnostic messages written to standard error.

28837 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

## 28838 **ASYNCHRONOUS EVENTS**

28839 Default.

## 28840 **STDOUT**

28841 The standard output shall be a text file whose format is dependent on the data keywords  
28842 specified with the **-d** option.

### 28843 **Data Keywords**

28844 Data keywords specify which parts of an SCCS file shall be retrieved and output. All parts of an  
28845 SCCS file have an associated data keyword. A data keyword may appear in a *dataspec* multiple  
28846 times.

28847 The information written by *prs* shall consist of:

- 28848 1. The user-supplied text
- 28849 2. Appropriate values (extracted from the SCCS file) substituted for the recognized data  
28850 keywords in the order of appearance in the *dataspec*

28851 The format of a data keyword value shall either be simple (‘S’), in which keyword substitution  
28852 is direct, or multi-line (‘M’).

28853 User-supplied text shall be any text other than recognized data keywords. A **<tab>** shall be  
28854 specified by ‘\t’ and **<newline>** by ‘\n’. When the **-r** option is not specified, the default  
28855 *dataspec* shall be:

28856 :PN: :\n\n

28857 and the following *dataspec* shall be used for each selected delta:

28858 :Dt:\t:DL:\nMRs:\n:MR:COMMENTS:\n:C:

28859

28860

28861

| SCCS File Data Keywords |                                                          |              |                |        |
|-------------------------|----------------------------------------------------------|--------------|----------------|--------|
| Keyword                 | Data Item                                                | File Section | Value          | Format |
| :Dt:                    | Delta information                                        | Delta Table  | See below*     | S      |
| :DL:                    | Delta line statistics                                    | "            | :Li:/:Ld:/:Lu: | S      |
| :Li:                    | Lines inserted by Delta                                  | "            | nnnnn***       | S      |
| :Ld:                    | Lines deleted by Delta                                   | "            | nnnnn***       | S      |
| :Lu:                    | Lines unchanged by Delta                                 | "            | nnnnn***       | S      |
| :DT:                    | Delta type                                               | "            | D or R         | S      |
| :I:                     | SCCS ID string (SID)                                     | "            | See below**    | S      |
| :R:                     | Release number                                           | "            | nnnn           | S      |
| :L:                     | Level number                                             | "            | nnnn           | S      |
| :B:                     | Branch number                                            | "            | nnnn           | S      |
| :S:                     | Sequence number                                          | "            | nnnn           | S      |
| :D:                     | Date delta created                                       | "            | :Dy:/:Dm:/:Dd: | S      |
| :Dy:                    | Year delta created                                       | "            | nn             | S      |
| :Dm:                    | Month delta created                                      | "            | nn             | S      |
| :Dd:                    | Day delta created                                        | "            | nn             | S      |
| :T:                     | Time delta created                                       | "            | :Th:::Tm:::Ts: | S      |
| :Th:                    | Hour delta created                                       | "            | nn             | S      |
| :Tm:                    | Minutes delta created                                    | "            | nn             | S      |
| :Ts:                    | Seconds delta created                                    | "            | nn             | S      |
| :P:                     | Programmer who created Delta                             | "            | logname        | S      |
| :DS:                    | Delta sequence number                                    | "            | nnnn           | S      |
| :DP:                    | Predecessor Delta sequence number                        | "            | nnnn           | S      |
| :DI:                    | Sequence number of deltas included, excluded, or ignored | "            | :Dn:/:Dx:/:Dg: | S      |
| :Dn:                    | Deltas included (sequence #)                             | "            | :DS: :DS: ...  | S      |
| :Dx:                    | Deltas excluded (sequence #)                             | "            | :DS: :DS: ...  | S      |
| :Dg:                    | Deltas ignored (sequence #)                              | "            | :DS: :DS: ...  | S      |
| :MR:                    | MR numbers for delta                                     | "            | text           | M      |
| :C:                     | Comments for delta                                       | "            | text           | M      |
| :UN:                    | User names                                               | User Names   | text           | M      |
| :FL:                    | Flag list                                                |              | Flags          | M      |
| :Y:                     | Module type flag                                         | "            | text           | S      |
| :MF:                    | MR validation flag                                       | "            | yes or no      | S      |
| :MP:                    | MR validation program name                               | "            | text           | S      |
| :KF:                    | Keyword error, warning flag                              | "            | yes or no      | S      |
| :KV:                    | Keyword validation string                                | "            | text           | S      |
| :BF:                    | Branch flag                                              | "            | yes or no      | S      |
| :J:                     | Joint edit flag                                          | "            | yes or no      | S      |
| :LK:                    | Locked releases                                          | "            | :R: ...        | S      |
| :Q:                     | User-defined keyword                                     | "            | text           | S      |
| :M:                     | Module name                                              | "            | text           | S      |

28904

28905

| SCCS File Data Keywords |                       |              |                 |        |
|-------------------------|-----------------------|--------------|-----------------|--------|
| Keyword                 | Data Item             | File Section | Value           | Format |
| :FB:                    | Floor boundary        | "            | :R:             | S      |
| :CB:                    | Ceiling boundary      | "            | :R:             | S      |
| :Ds:                    | Default SID           | "            | :I:             | S      |
| :ND:                    | Null delta flag       | "            | yes or no       | S      |
| :FD:                    | File descriptive text | Comments     | text            | M      |
| :BD:                    | Body                  | Body         | text            | M      |
| :GB:                    | Gotten body           | "            | text            | M      |
| :W:                     | A form of what string | N/A          | :Z::M:\t:I:     | S      |
| :A:                     | A form of what string | N/A          | :Z::Y::M::I::Z: | S      |
| :Z:                     | what string delimiter | N/A          | @( # )          | S      |
| :F:                     | SCCS filename         | N/A          | text            | S      |
| :PN:                    | SCCS file pathname    | N/A          | text            | S      |

28919

\* :Dt:=:DT: :I: :D: :T: :P: :DS: :DP:

28920

\*\* :R::L::B::S: if the delta is a branch delta (:BF: = yes)  
:R::L: if the delta is not a branch delta (:BF: = no)

28922

\*\*\* The line statistics are capped at 99 999. For example, if 100 000 lines were unchanged in a certain revision, :Lu: shall produce the value 99 999.

28923

**28924 STDERR**

28925

The standard error shall be used only for diagnostic messages.

28926 **OUTPUT FILES**

28927

None.

28928 **EXTENDED DESCRIPTION**

28929

None.

28930 **EXIT STATUS**

28931

The following exit values shall be returned:

28932

0 Successful completion.

28933

&gt;0 An error occurred.

28934 **CONSEQUENCES OF ERRORS**

28935

Default.

28936 **APPLICATION USAGE**

28937

None.

28938 **EXAMPLES**

28939

1. The following example:

28940

prs -d "User Names for :F: are:\n:UN:" s.file

28941

might write to standard output:

28942

User Names for s.file are:

28943

xyz

28944

131

28945

abc

28946

2. The following example:

28947            prs -d "Delta for pgm :M:: :I: - :D: By :P:" -r s.file  
28948            might write to standard output:  
28949            Delta for pgm main.c: 3.7 - 77/12/01 By cas  
28950            3. As a special case:  
28951            prs s.file  
28952            might write to standard output:  
28953            s.file:  
28954            <blank line>  
28955            D 1.1 77/12/01 00:00:00 cas 1 000000/00000/00000  
28956            MRS:  
28957            bl78-12345  
28958            bl79-54321  
28959            COMMENTS:  
28960            this is the comment line for s.file initial delta  
28961            <blank line>  
28962            for each delta table entry of the **D** type. The only option allowed to be used with this  
28963            special case is the **-a** option.

#### 28964 RATIONALE

28965            None.

#### 28966 FUTURE DIRECTIONS

28967            None.

#### 28968 SEE ALSO

28969            *admin, delta, get, what*

#### 28970 CHANGE HISTORY

28971            First released in Issue 2.

#### 28972 Issue 5

28973            The phrase “in which keyword substitution is followed by a <newline>” is deleted from the end  
28974            of the second paragraph of **Data Keywords** (on page 744).

28975            The interpretation of the YY component of the **-c cutoff** argument is noted.

#### 28976 Issue 6

28977            The normative text is reworded to emphasize the term “shall” for implementation requirements.

28978            The Open Group Base Resolution bwg2001-007 is applied, updating the table in STDOUT with a  
28979            note that line statistics are capped at 99 999 for the **:Li:**, **:Ld:**, **:Lu:**, and **:DL:** keywords.

28980            The Open Group Interpretation PIN4C.00009 is applied.

## 28981 NAME

28982 ps — report process status

## 28983 SYNOPSIS

28984 UP XSI ps [-aA][-defl][-G *grouplist*] [-o *format*]... [-p *proclist*] [-t *termlist*]  
28985 [-U *userlist*] [-g *grouplist*] [-n *namelist*] [-u *userlist*]

28986

## 28987 DESCRIPTION

28988 The *ps* utility shall write information about processes, subject to having the appropriate  
28989 privileges to obtain information about those processes.28990 By default, *ps* shall select all processes with the same effective user ID as the current user and the  
28991 same controlling terminal as the invoker.

## 28992 OPTIONS

28993 The *ps* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
28994 Utility Syntax Guidelines.

28995 The following options shall be supported:

28996 -a Write information for all processes associated with terminals. Implementations  
28997 may omit session leaders from this list.

28998 -A Write information for all processes.

28999 XSI -d Write information for all processes, except session leaders.

29000 XSI -e Write information for all processes. (Equivalent to -A.)

29001 XSI -f Generate a **full** listing. (See the STDOUT section for the contents of a **full** listing.)29002 XSI -g *grouplist* Write information for processes whose session leaders are given in *grouplist*. The  
29003 application shall ensure that the *grouplist* is a single argument in the form of a  
29004 <blank> or comma-separated list.29005 -G *grouplist* Write information for processes whose real group ID numbers are given in *grouplist*. The  
29006 application shall ensure that the *grouplist* is a single argument in the form of a  
29007 <blank> or comma-separated list.29008 XSI -l Generate a **long** listing. (See STDOUT for the contents of a **long** listing.)29009 XSI -n *namelist* Specify the name of an alternative system *namelist* file in place of the default. The  
29010 name of the default file and the format of a *namelist* file are unspecified.29011 -o *format* Write information according to the format specification given in *format*. This is  
29012 fully described in the STDOUT section. Multiple -o options can be specified; the  
29013 format specification shall be interpreted as the <space>-separated concatenation of  
29014 all the *format* option-arguments.29015 -p *proclist* Write information for processes whose process ID numbers are given in *proclist*.  
29016 The application shall ensure that the *proclist* is a single argument in the form of a  
29017 <blank> or comma-separated list.29018 -t *termlist* Write information for processes associated with terminals given in *termlist*. The  
29019 application shall ensure that the *termlist* is a single argument in the form of a  
29020 <blank> or comma-separated list. Terminal identifiers shall be given in an  
29021 XSI implementation-defined format. On XSI-conformant systems, they shall be given  
29022 in one of two forms: the device's filename (for example, **tty04**) or, if the device's  
29023 filename starts with **tty**, just the identifier following the characters **tty** (for

|           |                              |                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29024     |                              | example, "0 4").                                                                                                                                                                                                                                                                                                                                                            |
| 29025 XSI | <b>-u userlist</b>           | Write information for processes whose user ID numbers or login names are given in <i>userlist</i> . The application shall ensure that the <i>userlist</i> is a single argument in the form of a <blank> or comma-separated list. In the listing, the numerical user ID shall be written unless the <b>-f</b> option is used, in which case the login name shall be written. |
| 29026     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29027     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29028     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29029     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29030     | <b>-U userlist</b>           | Write information for processes whose real user ID numbers or login names are given in <i>userlist</i> . The application shall ensure that the <i>userlist</i> is a single argument in the form of a <blank> or comma-separated list.                                                                                                                                       |
| 29031     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29032     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29033     |                              | With the exception of <b>-o format</b> , all of the options shown are used to select processes. If any are specified, the default list shall be ignored and <i>ps</i> shall select the processes represented by the inclusive OR of all the selection-criteria options.                                                                                                     |
| 29034     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29035     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29036     | <b>OPERANDS</b>              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29037     |                              | None.                                                                                                                                                                                                                                                                                                                                                                       |
| 29038     | <b>STDIN</b>                 |                                                                                                                                                                                                                                                                                                                                                                             |
| 29039     |                              | Not used.                                                                                                                                                                                                                                                                                                                                                                   |
| 29040     | <b>INPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                             |
| 29041     |                              | None.                                                                                                                                                                                                                                                                                                                                                                       |
| 29042     | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                                                                                                                                                                                                                                                                             |
| 29043     |                              | The following environment variables shall affect the execution of <i>ps</i> :                                                                                                                                                                                                                                                                                               |
| 29044     | <b>COLUMNS</b>               | Override the system-selected horizontal display line size, used to determine the number of text columns to display. See the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables for valid values and results when it is unset or null.                                                                                                        |
| 29045     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29046     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29047     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29048     | <b>LANG</b>                  | Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)                                                                       |
| 29049     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29050     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29051     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29052     | <b>LC_ALL</b>                | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                    |
| 29053     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29054     | <b>LC_CTYPE</b>              | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                                   |
| 29055     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29056     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29057     | <b>LC_MESSAGES</b>           | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.                                                                                                                                                                                            |
| 29058     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29059     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29060     |                              |                                                                                                                                                                                                                                                                                                                                                                             |
| 29061     | <b>LC_TIME</b>               | Determine the format and contents of the date and time strings displayed.                                                                                                                                                                                                                                                                                                   |
| 29062 XSI | <b>NLSPATH</b>               | Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                                                                                                                                                                                                                                                                       |
| 29063     | <b>TZ</b>                    | Determine the timezone used to calculate date and time strings displayed. If <b>TZ</b> is unset or null, an unspecified default timezone shall be used.                                                                                                                                                                                                                     |
| 29064     |                              |                                                                                                                                                                                                                                                                                                                                                                             |

## 29065 ASYNCHRONOUS EVENTS

29066 Default.

## 29067 STDOUT

29068 When the **-o** option is not specified, the standard output format is unspecified.

29069 XSI On XSI-conformant systems, the output format shall be as follows. The column headings and descriptions of the columns in a *ps* listing are given below. The precise meanings of these fields are implementation-defined. The letters '**f**' and '**l**' (below) indicate the option (**full** or **long**) that shall cause the corresponding heading to appear; **all** means that the heading always appears. Note that these two options determine only what information is provided for a process; they do not determine which processes are listed.

|       |              |       |                                                                                                   |
|-------|--------------|-------|---------------------------------------------------------------------------------------------------|
| 29075 | <b>F</b>     | (l)   | Flags (octal and additive) associated with the process.                                           |
| 29076 | <b>S</b>     | (l)   | The state of the process.                                                                         |
| 29077 | <b>UID</b>   | (f,l) | The user ID number of the process owner; the login name is printed under the <b>-f</b> option.    |
| 29078 | <b>PID</b>   | (all) | The process ID of the process; it is possible to kill a process if this datum is known.           |
| 29079 | <b>PPID</b>  | (f,l) | The process ID of the parent process.                                                             |
| 29082 | <b>C</b>     | (f,l) | Processor utilization for scheduling.                                                             |
| 29083 | <b>PRI</b>   | (l)   | The priority of the process; higher numbers mean lower priority.                                  |
| 29084 | <b>NI</b>    | (l)   | Nice value; used in priority computation.                                                         |
| 29085 | <b>ADDR</b>  | (l)   | The address of the process.                                                                       |
| 29086 | <b>SZ</b>    | (l)   | The size in blocks of the core image of the process.                                              |
| 29087 | <b>WCHAN</b> | (l)   | The event for which the process is waiting or sleeping; if blank, the process is running.         |
| 29089 | <b>STIME</b> | (f)   | Starting time of the process.                                                                     |
| 29090 | <b>TTY</b>   | (all) | The controlling terminal for the process.                                                         |
| 29091 | <b>TIME</b>  | (all) | The cumulative execution time for the process.                                                    |
| 29092 | <b>CMD</b>   | (all) | The command name; the full command name and its arguments are written under the <b>-f</b> option. |
| 29093 |              |       |                                                                                                   |

29094 A process that has exited and has a parent, but has not yet been waited for by the parent, shall be marked **defunct**.

29096 Under the option **-f**, *ps* tries to determine the command name and arguments given when the process was created by examining memory or the swap area. Failing this, the command name, as it would appear without the option **-f**, is written in square brackets.

29099 The **-o** option allows the output format to be specified under user control.

29100 The application shall ensure that the format specification is a list of names presented as a single argument, <blank> or comma-separated. Each variable has a default header. The default header can be overridden by appending an equals sign and the new text of the header. The rest of the characters in the argument shall be used as the header text. The fields specified shall be written in the order specified on the command line, and should be arranged in columns in the output. The field widths shall be selected by the system to be at least as wide as the header text (default or overridden value). If the header text is null, such as **-o user=**, the field width shall be at least as wide as the default header text. If all header text fields are null, no header line shall be written.

29109 The following names are recognized in the POSIX locale:

|       |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29110 | <b>ruser</b>  | The real user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.                                                                                                                                                                                                                                                                                                                                                                                            |
| 29111 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29112 | <b>user</b>   | The effective user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.                                                                                                                                                                                                                                                                                                                                                                                       |
| 29113 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29114 | <b>rgroup</b> | The real group ID of the process. This shall be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.                                                                                                                                                                                                                                                                                                                                                                                          |
| 29115 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29116 | <b>group</b>  | The effective group ID of the process. This shall be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.                                                                                                                                                                                                                                                                                                                                                                                     |
| 29117 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29118 | <b>pid</b>    | The decimal value of the process ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 29119 | <b>ppid</b>   | The decimal value of the parent process ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 29120 | <b>pgid</b>   | The decimal value of the process group ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 29121 | <b>pcpu</b>   | The ratio of CPU time used recently to CPU time available in the same period, expressed as a percentage. The meaning of “recently” in this context is unspecified. The CPU time available is determined in an unspecified manner.                                                                                                                                                                                                                                                                                                                        |
| 29122 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29123 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29124 | <b>vsz</b>    | The size of the process in (virtual) memory in 1 024 byte units as a decimal integer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 29125 | <b>nice</b>   | The decimal value of the nice value of the process; see <i>nice</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 29126 | <b>etime</b>  | In the POSIX locale, the elapsed time since the process was started, in the form:<br>[[dd-]hh:]mm:ss                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 29127 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29128 |               | where <i>dd</i> shall represent the number of days, <i>hh</i> the number of hours, <i>mm</i> the number of minutes, and <i>ss</i> the number of seconds. The <i>dd</i> field shall be a decimal integer. The <i>hh</i> , <i>mm</i> , and <i>ss</i> fields shall be two-digit decimal integers padded on the left with zeros.                                                                                                                                                                                                                             |
| 29129 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29130 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29131 | <b>time</b>   | In the POSIX locale, the cumulative CPU time of the process in the form:<br>[dd-]hh:mm:ss                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29132 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29133 |               | The <i>dd</i> , <i>hh</i> , <i>mm</i> , and <i>ss</i> fields shall be as described in the <b>etime</b> specifier.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 29134 | <b>tty</b>    | The name of the controlling terminal of the process (if any) in the same format used by the <i>who</i> utility.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29135 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29136 | <b>comm</b>   | The name of the command being executed ( <i>argv[0]</i> value) as a string.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 29137 | <b>args</b>   | The command with all its arguments as a string. The implementation may truncate this value to the field width; it is implementation-defined whether any further truncation occurs. It is unspecified whether the string represented is a version of the argument list as it was passed to the command when it started, or is a version of the arguments as they may have been modified by the application. Applications cannot depend on being able to modify their argument list and having that modification be reflected in the output of <i>ps</i> . |
| 29138 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29139 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29140 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29141 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29142 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29143 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29144 |               | Any field need not be meaningful in all implementations. In such a case a hyphen ('-') should be output in place of the field value.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 29145 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29146 |               | Only <b>comm</b> and <b>args</b> shall be allowed to contain <blank>s; all others shall not. Any implementation-defined variables shall be specified in the system documentation along with the default header and indicating whether the field may contain <blank>s.                                                                                                                                                                                                                                                                                    |
| 29147 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29148 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29149 |               | The following table specifies the default header to be used in the POSIX locale corresponding to each format specifier.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 29150 |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

29151

**Table 4-17** Variable Names and Default Headers in *ps*

| Format Specifier | Default Header | Format Specifier | Default Header |
|------------------|----------------|------------------|----------------|
| <b>args</b>      | COMMAND        | <b>ppid</b>      | PPID           |
| <b>comm</b>      | COMMAND        | <b>rgroup</b>    | RGROUP         |
| <b>etime</b>     | ELAPSED        | <b>ruser</b>     | RUSER          |
| <b>group</b>     | GROUP          | <b>time</b>      | TIME           |
| <b>nice</b>      | NI             | <b>tty</b>       | TT             |
| <b>pcpu</b>      | %CPU           | <b>user</b>      | USER           |
| <b>pgid</b>      | PGID           | <b>vsz</b>       | VSZ            |
| <b>pid</b>       | PID            |                  |                |

29161 **STDERR**

29162 The standard error shall be used only for diagnostic messages.

29163 **OUTPUT FILES**

29164 None.

29165 **EXTENDED DESCRIPTION**

29166 None.

29167 **EXIT STATUS**

29168 The following exit values shall be returned:

29169 0 Successful completion.

29170 &gt;0 An error occurred.

29171 **CONSEQUENCES OF ERRORS**

29172 Default.

29173 **APPLICATION USAGE**29174 Things can change while *ps* is running; the snapshot it gives is only true for an instant, and might  
29175 not be accurate by the time it is displayed.29176 The **args** format specifier is allowed to produce a truncated version of the command arguments.  
29177 In some implementations, this information is no longer available when the *ps* utility is executed.29178 If the field width is too narrow to display a textual ID, the system may use a numeric version.  
29179 Normally, the system would be expected to choose large enough field widths, but if a large  
29180 number of fields were selected to write, it might squeeze fields to their minimum sizes to fit on  
29181 one line. One way to ensure adequate width for the textual IDs is to override the default header  
29182 for a field to make it larger than most or all user or group names.29183 There is no special quoting mechanism for header text. The header text is the rest of the  
29184 argument. If multiple header changes are needed, multiple **-o** options can be used, such as:29185 *ps -o "user=User Name" -o pid=Process\ ID*29186 On some implementations, especially multi-level secure systems, *ps* may be severely restricted  
29187 and produce information only about child processes owned by the user.29188 **EXAMPLES**

29189 The command:

29190 *ps -o user,pid,ppid=MOM -o args*

29191 writes at least the following in the POSIX locale:

29192       USER    PID    MOM    COMMAND  
29193       helene   34     12    *ps -o uid,pid,ppid=MOM -o args*

29194        The contents of the **COMMAND** field need not be the same in all implementations, due to  
29195        possible truncation.

29196 **RATIONALE**

29197        There is very little commonality between BSD and System V implementations of *ps*. Many  
29198        options conflict or have subtly different usages. The standard developers attempted to select a  
29199        set of options for the base standard that were useful on a wide range of systems and selected  
29200        options that either can be implemented on both BSD and System V-based systems without  
29201        breaking the current implementations or where the options are sufficiently similar that any  
29202        changes would not be unduly problematic for users or implementors.

29203        It is recognized that on some implementations, especially multi-level secure systems, *ps* may be  
29204        nearly useless. The default output has therefore been chosen such that it does not break  
29205        historical implementations and also is likely to provide at least some useful information on most  
29206        systems.

29207        The major change is the addition of the format specification capability. The motivation for this  
29208        invention is to provide a mechanism for users to access a wider range of system information, if  
29209        the system permits it, in a portable manner. The fields chosen to appear in this volume of  
29210        IEEE Std 1003.1-2001 were arrived at after considering what concepts were likely to be both  
29211        reasonably useful to the “average” user and had a reasonable chance of being implemented on a  
29212        wide range of systems. Again it is recognized that not all systems are able to provide all the  
29213        information and, conversely, some may wish to provide more. It is hoped that the approach  
29214        adopted will be sufficiently flexible and extensible to accommodate most systems.  
29215        Implementations may be expected to introduce new format specifiers.

29216        The default output should consist of a short listing containing the process ID, terminal name,  
29217        cumulative execution time, and command name of each process.

29218        The preference of the standard developers would have been to make the format specification an  
29219        operand of the *ps* command. Unfortunately, BSD usage precluded this.

29220        At one time a format was included to display the environment array of the process. This was  
29221        deleted because there is no portable way to display it.

29222        The **-A** option is equivalent to the BSD **-g** and the SVID **-e**. Because the two systems differed, a  
29223        mnemonic compromise was selected.

29224        The **-a** option is described with some optional behavior because the SVID omits session leaders,  
29225        but BSD does not.

29226        In an early proposal, format specifiers appeared for priority and start time. The former was not  
29227        defined adequately in this volume of IEEE Std 1003.1-2001 and was removed in deference to the  
29228        defined nice value; the latter because elapsed time was considered to be more useful.

29229        In a new BSD version of *ps*, a **-O** option can be used to write all of the default information,  
29230        followed by additional format specifiers. This was not adopted because the default output is  
29231        implementation-defined. Nevertheless, this is a useful option that should be reserved for that  
29232        purpose. In the **-o** option for the POSIX Shell and Utilities *ps*, the format is the concatenation of  
29233        each **-o**. Therefore, the user can have an alias or function that defines the beginning of their  
29234        desired format and add more fields to the end of the output in certain cases where that would be  
29235        useful.

29236        The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
29237        require that they all use the same format.

29238        The **pcpu** field indicates that the CPU time available is determined in an unspecified manner.  
29239        This is because it is difficult to express an algorithm that is useful across all possible machine

29240 architectures. Historical counterparts to this value have attempted to show percentage of use in  
29241 the recent past, such as the preceding minute. Frequently, these values for all processes did not  
29242 add up to 100%. Implementations are encouraged to provide data in this field to users that will  
29243 help them identify processes currently affecting the performance of the system.

29244 **FUTURE DIRECTIONS**

29245 None.

29246 **SEE ALSO**

29247 *kill, nice, renice*

29248 **CHANGE HISTORY**

29249 First released in Issue 2.

29250 **Issue 6**

29251 This utility is marked as part of the User Portability Utilities option.

29252 The normative text is reworded to avoid use of the term “must” for application requirements.

29253 The *TZ* entry is added to the ENVIRONMENT VARIABLES section.

**29254 NAME**

29255        *pwd* — return working directory name

**29256 SYNOPSIS**

29257        *pwd* [−L | −P ]

**29258 DESCRIPTION**

29259        The *pwd* utility shall write to standard output an absolute pathname of the current working  
29260        directory, which does not contain the filenames dot or dot-dot.

**29261 OPTIONS**

29262        The *pwd* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
29263        12.2, Utility Syntax Guidelines.

29264        The following options shall be supported by the implementation:

29265        **−L**        If the *PWD* environment variable contains an absolute pathname of the current  
29266        directory that does not contain the filenames dot or dot-dot, *pwd* shall write this  
29267        pathname to standard output. Otherwise, the **−L** option shall behave as the **−P**  
29268        option.

29269        **−P**        The absolute pathname written shall not contain filenames that, in the context of  
29270        the pathname, refer to files of type symbolic link.

29271        If both **−L** and **−P** are specified, the last one shall apply. If neither **−L** nor **−P** is specified, the *pwd*  
29272        utility shall behave as if **−L** had been specified.

**29273 OPERANDS**

29274        None.

**29275 STDIN**

29276        Not used.

**29277 INPUT FILES**

29278        None.

**29279 ENVIRONMENT VARIABLES**

29280        The following environment variables shall affect the execution of *pwd*:

29281        **LANG**      Provide a default value for the internationalization variables that are unset or null.  
29282        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
29283        Internationalization Variables for the precedence of internationalization variables  
29284        used to determine the values of locale categories.)

29285        **LC\_ALL**     If set to a non-empty string value, override the values of all the other  
29286        internationalization variables.

**29287 LC\_MESSAGES**

29288        Determine the locale that should be used to affect the format and contents of  
29289        diagnostic messages written to standard error.

29290        **XSI NLSPATH**      Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

29291        **PWD**        If the **−P** option is in effect, this variable shall be set to an absolute pathname of the  
29292        current working directory that does not contain any components that specify  
29293        symbolic links, does not contain any components that are dot, and does not  
29294        contain any components that are dot-dot. If an application sets or unsets the value  
29295        of *PWD*, the behavior of *pwd* is unspecified.

**29296 ASYNCHRONOUS EVENTS**

29297 Default.

**29298 STDOUT**

29299 The *pwd* utility output is an absolute pathname of the current working directory:

29300 "%s\n", <directory pathname>

**29301 STDERR**

29302 The standard error shall be used only for diagnostic messages.

**29303 OUTPUT FILES**

29304 None.

**29305 EXTENDED DESCRIPTION**

29306 None.

**29307 EXIT STATUS**

29308 The following exit values shall be returned:

29309 0 Successful completion.

29310 >0 An error occurred.

**29311 CONSEQUENCES OF ERRORS**

29312 If an error is detected, output shall not be written to standard output, a diagnostic message shall  
29313 be written to standard error, and the exit status is not zero.

**29314 APPLICATION USAGE**

29315 None.

**29316 EXAMPLES**

29317 None.

**29318 RATIONALE**

29319 Some implementations have historically provided *pwd* as a shell special built-in command.

29320 In most utilities, if an error occurs, partial output may be written to standard output. This does  
29321 not happen in historical implementations of *pwd*. Because *pwd* is frequently used in historical  
29322 shell scripts without checking the exit status, it is important that the historical behavior is  
29323 required here; therefore, the CONSEQUENCES OF ERRORS section specifically disallows any  
29324 partial output being written to standard output.

**29325 FUTURE DIRECTIONS**

29326 None.

**29327 SEE ALSO**

29328 *cd*, the System Interfaces volume of IEEE Std 1003.1-2001, *getcwd()*

**29329 CHANGE HISTORY**

29330 First released in Issue 2.

**29331 Issue 6**

29332 The **-P** and **-L** options are added to describe actions relating to symbolic links as specified in the  
29333 IEEE P1003.2b draft standard.

## 29334 NAME

29335 qalter — alter batch job

## 29336 SYNOPSIS

```
29337 BE qalter [-a date_time][-A account_string][-c interval][-e path_name]
29338 [-h hold_list][-j join_list][-k keep_list][-l resource_list]
29339 [-m mail_options][-M mail_list][-N name][-o path_name]
29340 [-p priority][-r y|n][-S path_name_list][-u user_list]
29341 job_identifier ...
29342
```

## 29343 DESCRIPTION

29344 The attributes of a batch job are altered by a request to the batch server that manages the batch  
 29345 job. The *qalter* utility is a user-accessible batch client that requests the alteration of the attributes  
 29346 of one or more batch jobs.

29347 The *qalter* utility shall alter the attributes of those batch jobs, and only those batch jobs, for which  
 29348 a batch *job\_identifier* is presented to the utility.

29349 The *qalter* utility shall alter the attributes of batch jobs in the order in which the batch  
 29350 *job\_identifiers* are presented to the utility.

29351 If the *qalter* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
 29352 process the remaining batch *job\_identifiers*, if any.

29353 For each batch *job\_identifier* for which the *qalter* utility succeeds, each attribute of the identified  
 29354 batch job shall be altered as indicated by all the options presented to the utility.

29355 For each identified batch job for which the *qalter* utility fails, the utility shall not alter any  
 29356 attribute of the batch job.

29357 For each batch job that the *qalter* utility processes, the utility shall not modify any attribute other  
 29358 than those required by the options and option-arguments presented to the utility.

29359 The *qalter* utility shall alter batch jobs by sending a *Modify Job Request* to the batch server that  
 29360 manages each batch job. At the time the *qalter* utility exits, it shall have modified the batch job  
 29361 corresponding to each successfully processed batch *job\_identifier*. An attempt to alter the  
 29362 attributes of a batch job in the RUNNING state is implementation-defined.

## 29363 OPTIONS

29364 The *qalter* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 29365 12.2, Utility Syntax Guidelines.

29366 The following options shall be supported by the implementation:

29367 **-a date\_time** Redefine the time at which the batch job becomes eligible for execution.

29368 The *date\_time* argument shall be in the same form and represent the same time as  
 29369 for the *touch* utility. The time so represented shall be set into the *Execution\_Time*  
 29370 attribute of the batch job. If the time specified is earlier than the current time, the  
 29371 **-a** option shall have no effect.

29372 **-A account\_string**

29373 Redefine the account to which the resource consumption of the batch job should be  
 29374 charged.

29375 The syntax of the *account\_string* option-argument is unspecified.

29376 The *qalter* utility shall set the *Account\_Name* attribute of the batch job to the value  
 29377 of the *account\_string* option-argument.

- 29378       **-c interval** Redefine whether the batch job should be checkpointed, and if so, how often.
- 29379           The *qalter* utility shall accept a value for the *interval* option-argument that is one of
- 29380           the following:
- 29381            n        No checkpointing is to be performed on the batch job  
29382                   (NO\_CHECKPOINT).
- 29383            s        Checkpointing is to be performed only when the batch server is shut  
29384                   down (CHECKPOINT\_AT\_SHUTDOWN).
- 29385            c        Automatic periodic checkpointing is to be performed at the  
29386                   *Minimum\_Cpu\_Interval* attribute of the batch queue, in units of CPU  
29387                   minutes (CHECKPOINT\_AT\_MIN\_CPU\_INTERVAL).
- 29388            c=minutes Automatic periodic checkpointing is to be performed every *minutes*  
29389                   of CPU time, or every *Minimum\_Cpu\_Interval* minutes, whichever is  
29390                   greater. The *minutes* argument shall conform to the syntax for  
29391                   unsigned integers and shall be greater than zero.
- 29392           An implementation may define other checkpoint intervals. The conformance  
29393           document for an implementation shall describe any alternative checkpoint  
29394           intervals, how they are specified, their internal behavior, and how they affect the  
29395           behavior of the utility.
- 29396           The *qalter* utility shall set the *Checkpoint* attribute of the batch job to the value of the  
29397           *interval* option-argument.
- 29398        **-e path\_name** Redefine the path to be used for the standard error stream of the batch job.
- 29399           The *qalter* utility shall accept a *path\_name* option-argument that conforms to the  
29400           syntax of the *path\_name* element defined in the System Interfaces volume of  
29401           IEEE Std 1003.1-2001, which can be preceded by a host name element of the form  
29402           *hostname*:
- 29403           If the *path\_name* option-argument constitutes an absolute pathname, the *qalter*  
29404           utility shall set the *Error\_Path* attribute of the batch job to the value of the  
29405           *path\_name* option-argument, including the host name element, if present.
- 29406           If the *path\_name* option-argument constitutes a relative pathname and no host  
29407           name element is specified, the *qalter* utility shall set the *Error\_Path* attribute of the  
29408           batch job to the value of the absolute pathname derived by expanding the  
29409           *path\_name* option-argument relative to the current directory of the process that  
29410           executes the *qalter* utility.
- 29411           If the *path\_name* option-argument constitutes a relative pathname and a host name  
29412           element is specified, the *qalter* utility shall set the *Error\_Path* attribute of the batch  
29413           job to the value of the option-argument without expansion.
- 29414           If the *path\_name* option-argument does not include a host name element, the *qalter*  
29415           utility shall prefix the pathname in the *Error\_Path* attribute with *hostname*: where  
29416           *hostname* is the name of the host upon which the *qalter* utility is being executed.
- 29417        **-h hold\_list** Redefine the types of holds, if any, on the batch job. The *qalter -h* option shall  
29418           accept a value for the *hold\_list* option-argument that is a string of alphanumeric  
29419           characters in the portable character set.
- 29420           The *qalter* utility shall accept a value for the *hold\_list* option-argument that is a  
29421           string of one or more of the characters 'u', 's', or 'o', or the single character  
29422           'n'. For each unique character in the *hold\_list* option-argument, the *qalter* utility

29423 shall add a value to the *Hold\_Types* attribute of the batch job as follows, each  
29424 representing a different hold type:

- 29425    u    USER  
29426    s    SYSTEM  
29427    o    OPERATOR

29428 If any of these characters are duplicated in the *hold\_list* option-argument, the  
29429 duplicates shall be ignored. An existing *Hold\_Types* attribute can be cleared by the  
29430 hold type:

- 29431    n    NO\_HOLD

29432 The *qalter* utility shall consider it an error if any hold type other than 'n' is  
29433 combined with hold type 'n'. Strictly conforming applications shall not repeat  
29434 any of the characters 'u', 's', 'o', or 'n' within the *hold\_list* option-argument.  
29435 The *qalter* utility shall permit the repetition of characters, but shall not assign  
29436 additional meaning to the repeated characters. An implementation may define  
29437 other hold types. The conformance document for an implementation shall describe  
29438 any additional hold types, how they are specified, their internal behavior, and how  
29439 they affect the behavior of the utility.

29440    -j *join\_list* Redefine which streams of the batch job are to be merged. The *qalter* -j option shall  
29441 accept a value for the *join\_list* option-argument that is a string of alphanumeric  
29442 characters in the portable character set.

29443 The *qalter* utility shall accept a *join\_list* option-argument that consists of one or  
29444 more of the characters 'e' and 'o', or the single character 'n'.

29445 All of the other batch job output streams specified shall be merged into the output  
29446 stream represented by the character listed first in the *join\_list* option-argument.

29447 For each unique character in the *join\_list* option-argument, the *qalter* utility shall  
29448 add a value to the *Join\_Path* attribute of the batch job as follows, each representing  
29449 a different batch job stream to join:

- 29450    e    The standard error of the batch job (JOIN\_STD\_ERROR).  
29451    o    The standard output of the batch job (JOIN\_STD\_OUTPUT).

29452 An existing *Join\_Path* attribute can be cleared by the join type:

- 29453    n    NO\_JOIN

29454 If 'n' is specified, then no files are joined. The *qalter* utility shall consider it an  
29455 error if any join type other than 'n' is combined with join type 'n'.

29456 Strictly conforming applications shall not repeat any of the characters 'e', 'o', or  
29457 'n' within the *join\_list* option-argument. The *qalter* utility shall permit the  
29458 repetition of characters, but shall not assign additional meaning to the repeated  
29459 characters.

29460 An implementation may define other join types. The conformance document for an  
29461 implementation shall describe any additional batch job streams, how they are  
29462 specified, their internal behavior, and how they affect the behavior of the utility.

29463    -k *keep\_list* Redefine which output of the batch job to retain on the execution host.

29464 The *qalter* -k option shall accept a value for the *keep\_list* option-argument that is a  
29465 string of alphanumeric characters in the portable character set.

29466      The *qalter* utility shall accept a *keep\_list* option-argument that consists of one or  
29467      more of the characters 'e' and 'o', or the single character 'n'.

29468      For each unique character in the *keep\_list* option-argument, the *qalter* utility shall  
29469      add a value to the *Keep\_Files* attribute of the batch job as follows, each representing  
29470      a different batch job stream to keep:

- 29471      e    The standard error of the batch job (KEEP\_STD\_ERROR).
- 29472      o    The standard output of the batch job (KEEP\_STD\_OUTPUT).

29473      If both 'e' and 'o' are specified, then both files are retained. An existing  
29474      *Keep\_Files* attribute can be cleared by the keep type:

29475      n    NO\_KEEP

29476      If 'n' is specified, then no files are retained. The *qalter* utility shall consider it an  
29477      error if any keep type other than 'n' is combined with keep type 'n'.

29478      Strictly conforming applications shall not repeat any of the characters 'e', 'o', or  
29479      'n' within the *keep\_list* option-argument. The *qalter* utility shall permit the  
29480      repetition of characters, but shall not assign additional meaning to the repeated  
29481      characters. An implementation may define other keep types. The conformance  
29482      document for an implementation shall describe any additional keep types, how  
29483      they are specified, their internal behavior, and how they affect the behavior of the  
29484      utility.

29485      **-I resource\_list**

29486      Redefine the resources that are allowed or required by the batch job.

29487      The *qalter* utility shall accept a *resource\_list* option-argument that conforms to the  
29488      following syntax:

29489      `resource=value[ , ,resource=value,... ]`

29490      The *qalter* utility shall set one entry in the value of the *Resource\_List* attribute of the  
29491      batch job for each resource listed in the *resource\_list* option-argument.

29492      Because the list of supported resource names might vary by batch server, the *qalter*  
29493      utility shall rely on the batch server to validate the resource names and associated  
29494      values. See Section 3.3.3 (on page 123) for a means of removing *keyword=value* (and  
29495      *value@keyword*) pairs and other general rules for list-oriented batch job attributes.

29496      **-m mail\_options**

29497      Redefine the points in the execution of the batch job at which the batch server is to  
29498      send mail about a change in the state of the batch job.

29499      The *qalter -m* option shall accept a value for the *mail\_options* option-argument that  
29500      is a string of alphanumeric characters in the portable character set.

29501      The *qalter* utility shall accept a value for the *mail\_options* option-argument that is a  
29502      string of one or more of the characters 'e', 'b', and 'a', or the single character  
29503      'n'. For each unique character in the *mail\_options* option-argument, the *qalter*  
29504      utility shall add a value to the *Mail\_Users* attribute of the batch job as follows, each  
29505      representing a different time during the life of a batch job at which to send mail:

- 29506      e    MAIL\_AT\_EXIT
- 29507      b    MAIL\_AT\_BEGINNING

29508           a MAIL\_AT\_ABORT  
29509         If any of these characters are duplicated in the *mail\_options* option-argument, the  
29510         duplicates shall be ignored.  
29511         An existing *Mail\_Points* attribute can be cleared by the mail type:  
29512           n NO\_MAIL  
29513         If 'n' is specified, then mail is not sent. The *qalter* utility shall consider it an error  
29514         if any mail type other than 'n' is combined with mail type 'n'. Strictly  
29515         conforming applications shall not repeat any of the characters 'e', 'b', 'a', or  
29516         'n' within the *mail\_options* option-argument. The *qalter* utility shall permit the  
29517         repetition of characters but shall not assign additional meaning to the repeated  
29518         characters.  
29519         An implementation may define other mail types. The conformance document for  
29520         an implementation shall describe any additional mail types, how they are  
29521         specified, their internal behavior, and how they affect the behavior of the utility.  
29522         **-M mail\_list** Redefine the list of users to which the batch server that executes the batch job is to  
29523         send mail, if the batch server sends mail about the batch job.  
29524         The syntax of the *mail\_list* option-argument is unspecified. If the implementation  
29525         of the *qalter* utility uses a name service to locate users, the utility shall accept the  
29526         syntax used by the name service.  
29527         If the implementation of the *qalter* utility does not use a name service to locate  
29528         users, the implementation shall accept the following syntax for user names:  
29529           mail\_address[ , mail\_address, . . . ]  
29530         The interpretation of *mail\_address* is implementation-defined.  
29531         The *qalter* utility shall set the *Mail\_Users* attribute of the batch job to the value of  
29532         the *mail\_list* option-argument.  
29533         **-N name** Redefine the name of the batch job.  
29534         The *qalter* -N option shall accept a value for the *name* option-argument that is a  
29535         string of up to 15 alphanumeric characters in the portable character set where the  
29536         first character is alphabetic.  
29537         The syntax of the *name* option-argument is unspecified.  
29538         The *qalter* utility shall set the *Job\_Name* attribute of the batch job to the value of the  
29539         *name* option-argument.  
29540         **-o path\_name** Redefine the path for the standard output of the batch job.  
29541         The *qalter* utility shall accept a *path\_name* option-argument that conforms to the  
29542         syntax of the *path\_name* element defined in the System Interfaces volume of  
29543         IEEE Std 1003.1-2001, which can be preceded by a host name element of the form  
29544         *hostname*:  
29545         If the *path\_name* option-argument constitutes an absolute pathname, the *qalter*  
29546         utility shall set the *Output\_Path* attribute of the batch job to the value of the  
29547         *path\_name* option-argument.  
29548         If the *path\_name* option-argument constitutes a relative pathname and no host  
29549         name element is specified, the *qalter* utility shall set the *Output\_Path* attribute of the  
29550         batch job to the absolute pathname derived by expanding the *path\_name* option-

29551 argument relative to the current directory of the process that executes the *qalter*  
29552 utility.

29553 If the *path\_name* option-argument constitutes a relative pathname and a host name  
29554 element is specified, the *qalter* utility shall set the *Output\_Path* attribute of the batch  
29555 job to the value of the *path\_name* option-argument without any expansion of the  
29556 pathname.

29557 If the *path\_name* option-argument does not include a host name element, the *qalter*  
29558 utility shall prefix the pathname in the *Output\_Path* attribute with *hostname*:*, where*  
29559 *hostname* is the name of the host upon which the *qalter* utility is being executed.

29560 **-p priority** Redefine the priority of the batch job.  
29561 The *qalter* utility shall accept a value for the priority option-argument that  
29562 conforms to the syntax for signed decimal integers, and which is not less than  
29563 -1 024 and not greater than 1 023.

29564 The *qalter* utility shall set the *Priority* attribute of the batch job to the value of the  
29565 *priority* option-argument.

29566 **-r y | n** Redefine whether the batch job is rerunnable.  
29567 If the value of the option-argument is 'y', the *qalter* utility shall set the *Rerunable*  
29568 attribute of the batch job to TRUE.  
29569 If the value of the option-argument is 'n', the *qalter* utility shall set the *Rerunable*  
29570 attribute of the batch job to FALSE.  
29571 The *qalter* utility shall consider it an error if any character other than 'y' or 'n' is  
29572 specified in the option-argument.

29573 **-S path\_name\_list** Redefine the shell that interprets the script at the destination system.  
29574 The *qalter* utility shall accept a *path\_name\_list* option-argument that conforms to  
29575 the following syntax:  
29577 pathname[@host] [, pathname[@host], ...]  
29578 The *qalter* utility shall accept only one pathname that is missing a corresponding  
29579 host name. The *qalter* utility shall allow only one pathname per named host.  
29580 The *qalter* utility shall add a value to the *Shell\_Path\_List* attribute of the batch job  
29581 for each entry in the *path\_name\_list* option-argument. See Section 3.3.3 (on page  
29582 123) for a means of removing *keyword*=*value* (and *value*@*keyword*) pairs and other  
29583 general rules for list-oriented batch job attributes.

29584 **-u user\_list** Redefine the user name under which the batch job is to run at the destination  
29585 system.  
29586 The *qalter* utility shall accept a *user\_list* option-argument that conforms to the  
29587 following syntax:  
29588 username[@host] [, , username[@host], ...]  
29589 The *qalter* utility shall accept only one user name that is missing a corresponding  
29590 host name. The *qalter* utility shall accept only one user name per named host.  
29591 The *qalter* utility shall add a value to the *User\_List* attribute of the batch job for each  
29592 entry in the *user\_list* option-argument. See Section 3.3.3 (on page 123) for a means  
29593 of removing *keyword*=*value* (and *value*@*keyword*) pairs and other general rules for

29594 list-oriented batch job attributes.

## 29595 OPERANDS

29596 The *qalter* utility shall accept one or more operands that conform to the syntax for a batch  
29597 *job\_identifier* (see Section 3.3.1 (on page 122)).

## 29598 STDIN

29599 Not used.

## 29600 INPUT FILES

29601 None.

## 29602 ENVIRONMENT VARIABLES

29603 The following environment variables shall affect the execution of *qalter*:

29604 *LANG* Provide a default value for the internationalization variables that are unset or null.  
29605 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
29606 Internationalization Variables for the precedence of internationalization variables  
29607 used to determine the values of locale categories.)

29608 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
29609 internationalization variables.

29610 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
29611 characters (for example, single-byte as opposed to multi-byte characters in  
29612 arguments).

### 29613 *LC\_MESSAGES*

29614 Determine the locale that should be used to affect the format and contents of  
29615 diagnostic messages written to standard error.

29616 *LOGNAME* Determine the login name of the user.

29617 *TZ* Determine the timezone used to interpret the *date-time* option-argument. If *TZ* is  
29618 unset or null, an unspecified default timezone shall be used.

## 29619 ASYNCHRONOUS EVENTS

29620 Default.

## 29621 STDOUT

29622 None.

## 29623 STDERR

29624 The standard error shall be used only for diagnostic messages.

## 29625 OUTPUT FILES

29626 None.

## 29627 EXTENDED DESCRIPTION

29628 None.

## 29629 EXIT STATUS

29630 The following exit values shall be returned:

29631 0 Successful completion.

29632 >0 An error occurred.

## 29633 CONSEQUENCES OF ERRORS

29634 In addition to the default behavior, the *qalter* utility shall not be required to write a diagnostic  
29635 message to standard error when the error reply received from a batch server indicates that the  
29636 batch *job\_identifier* does not exist on the server. Whether or not the *qalter* utility attempts to  
29637 locate the batch job on other batch servers is implementation-defined.

## 29638 APPLICATION USAGE

29639 None.

## 29640 EXAMPLES

29641 None.

## 29642 RATIONALE

29643 The *qalter* utility allows users to change the attributes of a batch job.

29644 As a means of altering a queued job, the *qalter* utility is superior to deleting and requeuing the  
29645 batch job insofar as an altered job retains its place in the queue with some traditional selection  
29646 algorithms. In addition, the *qalter* utility is both shorter and simpler than a sequence of *qdel* and  
29647 *qsub* utilities.

29648 The result of an attempt on the part of a user to alter a batch job in a RUNNING state is  
29649 implementation-defined because a batch job in the RUNNING state will already have opened its  
29650 output files and otherwise performed any actions indicated by the options in effect at the time  
29651 the batch job began execution.

29652 The options processed by the *qalter* utility are identical to those of the *qsub* utility, with a few  
29653 exceptions: **-V**, **-v**, and **-q**. The **-V** and **-v** are inappropriate for the *qalter* utility, since they  
29654 capture potentially transient environment information from the submitting process. The **-q**  
29655 option would specify a new queue, which would largely negate the previously stated advantage  
29656 of using *qalter*; furthermore, the *qmove* utility provides a superior means of moving jobs.

29657 Each of the following paragraphs provides the rationale for a *qalter* option.

29658 Additional rationale concerning these options can be found in the rationale for the *qsub* utility.

29659 The **-a** option allows users to alter the date and time at which a batch job becomes eligible to  
29660 run.

29661 The **-A** option allows users to change the account that will be charged for the resources  
29662 consumed by the batch job. Support for the **-A** option is mandatory for conforming  
29663 implementations of *qalter*, even though support of accounting is optional for servers. Whether or  
29664 not to support accounting is left to the implementor of the server, but mandatory support of the  
29665 **-A** option assures users of a consistent interface and allows them to control accounting on  
29666 servers that support accounting.

29667 The **-c** option allows users to alter the checkpointing interval of a batch job. A checkpointing  
29668 system, which is not defined by IEEE Std 1003.1-2001, allows recovery of a batch job at the most  
29669 recent checkpoint in the event of a crash. Checkpointing is typically used for jobs that consume  
29670 expensive computing time or must meet a critical schedule. Users should be allowed to make  
29671 the tradeoff between the overhead of checkpointing and the risk to the timely completion of the  
29672 batch job; therefore, this volume of IEEE Std 1003.1-2001 provides the checkpointing interval  
29673 option. Support for checkpointing is optional for servers.

29674 The **-e** option allows users to alter the name and location of the standard error stream written by  
29675 a batch job. However, the path of the standard error stream is meaningless if the value of the  
29676 *Join\_Path* attribute of the batch job is TRUE.

29677 The **-h** option allows users to set the hold type in the *Hold\_Types* attribute of a batch job. The  
29678 *qhold* and *qrsl* utilities add or remove hold types to the *Hold\_Types* attribute, respectively. The **-h**

29679 option has been modified to allow for implementation-defined hold types.

29680 The **-j** option allows users to alter the decision to join (merge) the standard error stream of the  
29681 batch job with the standard output stream of the batch job.

29682 The **-l** option allows users to change the resource limits imposed on a batch job.

29683 The **-m** option allows users to modify the list of points in the life of a batch job at which the  
29684 designated users will receive mail notification.

29685 The **-M** option allows users to alter the list of users who will receive notification about events in  
29686 the life of a batch job.

29687 The **-N** option allows users to change the name of a batch job.

29688 The **-o** option allows users to alter the name and path to which the standard output stream of  
29689 the batch job will be written.

29690 The **-P** option allows users to modify the priority of a batch job. Support for priority is optional  
29691 for batch servers.

29692 The **-r** option allows users to alter the rerunability status of a batch job.

29693 The **-S** option allows users to change the name and location of the shell image that will be  
29694 invoked to interpret the script of the batch job. This option has been modified to allow a list of  
29695 shell name and locations associated with different hosts.

29696 The **-u** option allows users to change the user identifier under which the batch job will execute.

29697 The *job\_identifier* operand syntax is provided so that the user can differentiate between the  
29698 originating and destination (or executing) batch server. These may or may not be the same. The  
29699 *.server\_name* portion identifies the originating batch server, while the *@server* portion identifies  
29700 the destination batch server.

29701 Historically, the *qalter* utility has been a component of the Network Queuing System (NQS), the  
29702 existing practice from which this utility has been derived.

### 29703 FUTURE DIRECTIONS

29704 None.

### 29705 SEE ALSO

29706 Chapter 3 (on page 101), *qdel*, *qhold*, *qmove*, *qrls*, *qsub*, *touch*

### 29707 CHANGE HISTORY

29708 Derived from IEEE Std 1003.2d-1994.

### 29709 Issue 6

29710 The *TZ* entry is added to the ENVIRONMENT VARIABLES section.

29711 IEEE PASC Interpretation 1003.2 #182 is applied, clarifying the description of the **-a** option.

**29712 NAME**

29713 qdel — delete batch jobs

**29714 SYNOPSIS**

29715 BE qdel *job\_identifier* ...

29716

**29717 DESCRIPTION**

29718 A batch job is deleted by sending a request to the batch server that manages the batch job. A  
29719 batch job that has been deleted is no longer subject to management by batch services.

29720 The *qdel* utility is a user-accessible client of batch services that requests the deletion of one or  
29721 more batch jobs.

29722 The *qdel* utility shall request a batch server to delete those batch jobs for which a batch  
29723 *job\_identifier* is presented to the utility.

29724 The *qdel* utility shall delete batch jobs in the order in which their batch *job\_identifiers* are  
29725 presented to the utility.

29726 If the *qdel* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed to  
29727 process the remaining batch *job\_identifiers*, if any.

29728 The *qdel* utility shall delete each batch job by sending a *Delete Job Request* to the batch server that  
29729 manages the batch job.

29730 The *qdel* utility shall not exit until the batch job corresponding to each successfully processed  
29731 batch *job\_identifier* has been deleted.

**29732 OPTIONS**

29733 None.

**29734 OPERANDS**

29735 The *qdel* utility shall accept one or more operands that conform to the syntax for a batch  
29736 *job\_identifier* (see Section 3.3.1 (on page 122)).

**29737 STDIN**

29738 Not used.

**29739 INPUT FILES**

29740 None.

**29741 ENVIRONMENT VARIABLES**

29742 The following environment variables shall affect the execution of *qdel*:

29743 *LANG* Provide a default value for the internationalization variables that are unset or null.  
29744 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
29745 Internationalization Variables for the precedence of internationalization variables  
29746 used to determine the values of locale categories.)

29747 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
29748 internationalization variables.

29749 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
29750 characters (for example, single-byte as opposed to multi-byte characters in  
29751 arguments).

**29752 *LC\_MESSAGES***

29753 Determine the locale that should be used to affect the format and contents of  
29754 diagnostic messages written to standard error.

- 29755        **LOGNAME** Determine the login name of the user.
- 29756 **ASYNCHRONOUS EVENTS**  
29757        Default.
- 29758 **STDOUT**  
29759        An implementation of the *qdel* utility may write informative messages to standard output.
- 29760 **STDERR**  
29761        The standard error shall be used only for diagnostic messages.
- 29762 **OUTPUT FILES**  
29763        None.
- 29764 **EXTENDED DESCRIPTION**  
29765        None.
- 29766 **EXIT STATUS**  
29767        The following exit values shall be returned:
- 29768        0 Successful completion.
- 29769        >0 An error occurred.
- 29770 **CONSEQUENCES OF ERRORS**  
29771        In addition to the default behavior, the *qdel* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qdel* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-defined.
- 29776 **APPLICATION USAGE**  
29777        None.
- 29778 **EXAMPLES**  
29779        None.
- 29780 **RATIONALE**  
29781        The *qdel* utility allows users and administrators to delete jobs.  
29782        The *qdel* utility provides functionality that is not otherwise available. For example, the *kill* utility of the operating system does not suffice. First, to use the *kill* utility, the user might have to log in on a remote node, because the *kill* utility does not operate across the network. Second, unlike *qdel*, *kill* cannot remove jobs from queues. Lastly, the arguments of the *qdel* utility are job identifiers rather than process identifiers, and so this utility can be passed the output of the *qselect* utility, thus providing users with a means of deleting a list of jobs.  
29788        Because a set of jobs can be selected using the *qselect* utility, the *qdel* utility has not been complicated with options that provide for selection of jobs. Instead, the batch jobs to be deleted are identified individually by their job identifiers.  
29791        Historically, the *qdel* utility has been a component of NQS, the existing practice on which it is based. However, the *qdel* utility defined in this volume of IEEE Std 1003.1-2001 does not provide an option for specifying a signal number to send to the batch job prior to the killing of the process; that capability has been subsumed by the *qsig* utility.  
29795        A discussion was held about the delays of networking and the possibility that the batch server may never respond, due to a down router, down batch server, or other network mishap. The DESCRIPTION records this under the words “fails to process any job identifier”. In the broad sense, the network problem is also an error, which causes the failure to process the batch job

29799 identifier.

**29800 FUTURE DIRECTIONS**

29801 None.

**29802 SEE ALSO**

29803 Chapter 3 (on page 101), *kill*, *qselect*, *qsig*

**29804 CHANGE HISTORY**

29805 Derived from IEEE Std 1003.2d-1994.

**29806 Issue 6**

29807 The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

## 29808 NAME

29809 qhold — hold batch jobs

## 29810 SYNOPSIS

29811 BE qhold [-h *hold\_list*] *job\_identifier* ...

29812

## 29813 DESCRIPTION

29814 A hold is placed on a batch job by a request to the batch server that manages the batch job. A  
29815 batch job that has one or more holds is not eligible for execution. The *qhold* utility is a user-  
29816 accessible client of batch services that requests one or more types of hold to be placed on one or  
29817 more batch jobs.

29818 The *qhold* utility shall place holds on those batch jobs for which a batch *job\_identifier* is presented  
29819 to the utility.

29820 The *qhold* utility shall place holds on batch jobs in the order in which their batch *job\_identifiers*  
29821 are presented to the utility. If the *qhold* utility fails to process any batch *job\_identifier* successfully,  
29822 the utility shall proceed to process the remaining batch *job\_identifiers*, if any.

29823 The *qhold* utility shall place holds on each batch job by sending a *Hold Job Request* to the batch  
29824 server that manages the batch job.

29825 The *qhold* utility shall not exit until holds have been placed on the batch job corresponding to  
29826 each successfully processed batch *job\_identifier*.

## 29827 OPTIONS

29828 The *qhold* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
29829 12.2, Utility Syntax Guidelines.

29830 The following option shall be supported by the implementation:

29831 **-h *hold\_list*** Define the types of holds to be placed on the batch job.

29832 The *qhold* **-h** option shall accept a value for the *hold\_list* option-argument that is a  
29833 string of alphanumeric characters in the portable character set (see the Base  
29834 Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

29835 The *qhold* utility shall accept a value for the *hold\_list* option-argument that is a  
29836 string of one or more of the characters 'u', 's', or 'o', or the single character  
29837 'n'.

29838 For each unique character in the *hold\_list* option-argument, the *qhold* utility shall  
29839 add a value to the *Hold\_Types* attribute of the batch job as follows, each  
29840 representing a different hold type:

29841    u    USER

29842    s    SYSTEM

29843    o    OPERATOR

29844 If any of these characters are duplicated in the *hold\_list* option-argument, the  
29845 duplicates shall be ignored.

29846 An existing *Hold\_Types* attribute can be cleared by the following hold type:

29847    n    NO\_HOLD

29848 The *qhold* utility shall consider it an error if any hold type other than 'n' is  
29849 combined with hold type 'n'.

29850 Strictly conforming applications shall not repeat any of the characters 'u', 's',  
29851 'o', or 'n' within the *hold\_list* option-argument. The *qhold* utility shall permit the  
29852 repetition of characters, but shall not assign additional meaning to the repeated  
29853 characters.

29854 An implementation may define other hold types. The conformance document for  
29855 an implementation shall describe any additional hold types, how they are  
29856 specified, their internal behavior, and how they affect the behavior of the utility.

29857 If the **-h** option is not presented to the *qhold* utility, the implementation shall set  
29858 the *Hold\_Types* attribute to USER.

## 29859 OPERANDS

29860 The *qhold* utility shall accept one or more operands that conform to the syntax for a batch  
29861 *job\_identifier* (see Section 3.3.1 (on page 122)).

## 29862 STDIN

29863 Not used.

## 29864 INPUT FILES

29865 None.

## 29866 ENVIRONMENT VARIABLES

29867 The following environment variables shall affect the execution of *qhold*:

29868 *LANG* Provide a default value for the internationalization variables that are unset or null.  
29869 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
29870 Internationalization Variables for the precedence of internationalization variables  
29871 used to determine the values of locale categories.)

29872 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
29873 internationalization variables.

29874 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
29875 characters (for example, single-byte as opposed to multi-byte characters in  
29876 arguments).

## 29877 *LC\_MESSAGES*

29878 Determine the locale that should be used to affect the format and contents of  
29879 diagnostic messages written to standard error.

29880 *LOGNAME* Determine the login name of the user.

## 29881 ASYNCHRONOUS EVENTS

29882 Default.

## 29883 STDOUT

29884 None.

## 29885 STDERR

29886 The standard error shall be used only for diagnostic messages.

## 29887 OUTPUT FILES

29888 None.

## 29889 EXTENDED DESCRIPTION

29890 None.

29891 **EXIT STATUS**

29892       The following exit values shall be returned:  
29893           0   Successful completion.  
29894           >0   An error occurred.

29895 **CONSEQUENCES OF ERRORS**

29896       In addition to the default behavior, the *qhold* utility shall not be required to write a diagnostic  
29897       message to standard error when the error reply received from a batch server indicates that the  
29898       batch *job\_identifier* does not exist on the server. Whether or not the *qhold* utility waits to output  
29899       the diagnostic message while attempting to locate the job on other servers is implementation-  
29900       defined.

29901 **APPLICATION USAGE**

29902       None.

29903 **EXAMPLES**

29904       None.

29905 **RATIONALE**

29906       The *qhold* utility allows users to place a hold on one or more jobs. A hold makes a batch job  
29907       ineligible for execution.

29908       The *qhold* utility has options that allow the user to specify the type of hold. Should the user wish  
29909       to place a hold on a set of jobs that meet a selection criteria, such a list of jobs can be acquired  
29910       using the *qselect* utility.

29911       The **-h** option allows the user to specify the type of hold that is to be placed on the job. This  
29912       option allows for USER, SYSTEM, OPERATOR, and implementation-defined hold types. The  
29913       USER and OPERATOR holds are distinct. The batch server that manages the batch job will verify  
29914       that the user is authorized to set the specified hold for the batch job.

29915       Mail is not required on hold because the administrator has the tools and libraries to build this  
29916       option if he or she wishes.

29917       Historically, the *qhold* utility has been a part of some existing batch systems, although it has not  
29918       traditionally been a part of the NQS.

29919 **FUTURE DIRECTIONS**

29920       None.

29921 **SEE ALSO**

29922       Chapter 3 (on page 101), *qselect*

29923 **CHANGE HISTORY**

29924       Derived from IEEE Std 1003.2d-1994.

29925 **Issue 6**

29926       The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

**29927 NAME**

29928 qmove — move batch jobs

**29929 SYNOPSIS**

29930 BE qmove destination job\_identifier ...

29931

**29932 DESCRIPTION**

29933 To move a batch job is to remove the batch job from the batch queue in which it resides and  
29934 instantiate the batch job in another batch queue. A batch job is moved by a request to the batch  
29935 server that manages the batch job. The *qmove* utility is a user-accessible batch client that requests  
29936 the movement of one or more batch jobs.

29937 The *qmove* utility shall move those batch jobs, and only those batch jobs, for which a batch  
29938 *job\_identifier* is presented to the utility.

29939 The *qmove* utility shall move batch jobs in the order in which the corresponding batch  
29940 *job\_identifiers* are presented to the utility.

29941 If the *qmove* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
29942 process the remaining batch *job\_identifiers*, if any.

29943 The *qmove* utility shall move batch jobs by sending a *Move Job Request* to the batch server that  
29944 manages each batch job. The *qmove* utility shall not exit before the batch jobs corresponding to all  
29945 successfully processed batch *job\_identifiers* have been moved.

**29946 OPTIONS**

29947 None.

**29948 OPERANDS**

29949 The *qmove* utility shall accept one operand that conforms to the syntax for a destination (see  
29950 Section 3.3.2 (on page 123)).

29951 The *qmove* utility shall accept one or more operands that conform to the syntax for a batch  
29952 *job\_identifier* (see Section 3.3.1 (on page 122)).

**29953 STDIN**

29954 Not used.

**29955 INPUT FILES**

29956 None.

**29957 ENVIRONMENT VARIABLES**

29958 The following environment variables shall affect the execution of *qmove*:

29959 *LANG* Provide a default value for the internationalization variables that are unset or null.  
29960 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
29961 Internationalization Variables for the precedence of internationalization variables  
29962 used to determine the values of locale categories.)

29963 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
29964 internationalization variables.

29965 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
29966 characters (for example, single-byte as opposed to multi-byte characters in  
29967 arguments).

**29968 *LC\_MESSAGES***

29969 Determine the locale that should be used to affect the format and contents of  
29970 diagnostic messages written to standard error.

29971       **LOGNAME** Determine the login name of the user.

29972 **ASYNCHRONOUS EVENTS**

29973       Default.

29974 **STDOUT**

29975       None.

29976 **STDERR**

29977       The standard error shall be used only for diagnostic messages.

29978 **OUTPUT FILES**

29979       None.

29980 **EXTENDED DESCRIPTION**

29981       None.

29982 **EXIT STATUS**

29983       The following exit values shall be returned:

29984       0   Successful completion.

29985       >0   An error occurred.

29986 **CONSEQUENCES OF ERRORS**

29987       In addition to the default behavior, the *qmove* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qmove* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-defined.

29992 **APPLICATION USAGE**

29993       None.

29994 **EXAMPLES**

29995       None.

29996 **RATIONALE**

29997       The *qmove* utility allows users to move jobs between queues.

29998       The alternative to using the *qmove* utility—deleting the batch job and requeueing it—entails  
29999       considerably more typing.

30000       Since the means of selecting jobs based on attributes has been encapsulated in the *qselect* utility,  
30001       the only option of the *qmove* utility concerns authorization. The **-u** option provides the user with  
30002       the convenience of changing the user identifier under which the batch job will execute.  
30003       Minimalism and consistency have taken precedence over convenience; the **-u** option has been  
30004       deleted because the equivalent capability exists with the **-u** option of the *qalter* utility.

30005 **FUTURE DIRECTIONS**

30006       None.

30007 **SEE ALSO**

30008       Chapter 3 (on page 101), *qalter*, *qselect*

30009 **CHANGE HISTORY**

30010       Derived from IEEE Std 1003.2d-1994.

**30011 Issue 6**

30012 The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

## 30013 NAME

30014 qmsg — send message to batch jobs

## 30015 SYNOPSIS

30016 BE qmsg [-E][-O] *message\_string job\_identifier ...*

30017

## 30018 DESCRIPTION

30019 To send a message to a batch job is to request that a server write a message string into one or  
30020 more output files of the batch job. A message is sent to a batch job by a request to the batch  
30021 server that manages the batch job. The *qmsg* utility is a user-accessible batch client that requests  
30022 the sending of messages to one or more batch jobs.

30023 The *qmsg* utility shall write messages into the files of batch jobs by sending a *Job Message Request*  
30024 to the batch server that manages the batch job. The *qmsg* utility shall not directly write the  
30025 message into the files of the batch job.

30026 The *qmsg* utility shall send a *Job Message Request* for those batch jobs, and only those batch jobs,  
30027 for which a batch *job\_identifier* is presented to the utility.

30028 The *qmsg* utility shall send *Job Message Requests* for batch jobs in the order in which their batch  
30029 *job\_identifiers* are presented to the utility.

30030 If the *qmsg* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed to  
30031 process the remaining batch *job\_identifiers*, if any.

30032 The *qmsg* utility shall not exit before a *Job Message Request* has been sent to the server that  
30033 manages the batch job that corresponds to each successfully processed batch *job\_identifier*.

## 30034 OPTIONS

30035 The *qmsg* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
30036 12.2, Utility Syntax Guidelines.

30037 The following options shall be supported by the implementation:

30038 -E Specify that the message is written to the standard error of each batch job.

30039 The *qmsg* utility shall write the message into the standard error of the batch job.

30040 -O Specify that the message is written to the standard output of each batch job.

30041 The *qmsg* utility shall write the message into the standard output of the batch job.

30042 If neither the -O nor the -E option is presented to the *qmsg* utility, the utility shall write the  
30043 message into an implementation-defined file. The conformance document for the  
30044 implementation shall describe the name and location of the implementation-defined file. If both  
30045 the -O and the -E options are presented to the *qmsg* utility, then the utility shall write the  
30046 messages to both standard output and standard error.

## 30047 OPERANDS

30048 The *qmsg* utility shall accept a minimum of two operands, *message\_string* and one or more batch  
30049 *job\_identifiers*.

30050 The *message\_string* operand shall be the string to be written to one or more output files of the  
30051 batch job followed by a <newline>. If the string contains <blank>s, then the application shall  
30052 ensure that the string is quoted. The *message\_string* shall be encoded in the portable character set  
30053 (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

30054 All remaining operands are batch *job\_identifiers* that conform to the syntax for a batch  
30055 *job\_identifier* (see Section 3.3.1 (on page 122)).

**30056 STDIN**

30057 Not used.

**30058 INPUT FILES**

30059 None.

**30060 ENVIRONMENT VARIABLES**

30061 The following environment variables shall affect the execution of *qmsg*:

30062 *LANG* Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

30066 *LC\_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

30068 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

**30071 *LC\_MESSAGES***

30072 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

30074 *LOGNAME* Determine the login name of the user.

**30075 ASYNCHRONOUS EVENTS**

30076 Default.

**30077 STDOUT**

30078 None.

**30079 STDERR**

30080 The standard error shall be used only for diagnostic messages.

**30081 OUTPUT FILES**

30082 None.

**30083 EXTENDED DESCRIPTION**

30084 None.

**30085 EXIT STATUS**

30086 The following exit values shall be returned:

30087 0 Successful completion.

30088 >0 An error occurred.

**30089 CONSEQUENCES OF ERRORS**

30090 In addition to the default behavior, the *qmsg* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qmsg* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-defined.

**30095 APPLICATION USAGE**

30096 None.

**30097 EXAMPLES**

30098 None.

**30099 RATIONALE**

30100 The *qmsg* utility allows users to write messages into the output files of running jobs. Users,  
30101 including operators and administrators, have a number of occasions when they want to place  
30102 messages in the output files of a batch job. For example, if a disk that is being used by a batch job  
30103 is showing errors, the operator might note this in the standard error stream of the batch job.

30104 The options of the *qmsg* utility provide users with the means of placing the message in the  
30105 output stream of their choice. The default output stream for the message—if the user does not  
30106 designate an output stream—is implementation-defined, since many implementations will  
30107 provide, as an extension to this volume of IEEE Std 1003.1-2001, a log file that shows the history  
30108 of utility execution.

30109 If users wish to send a message to a set of jobs that meet a selection criteria, the *qselect* utility can  
30110 be used to acquire the appropriate list of job identifiers.

30111 The **–E** option allows users to place the message in the standard error stream of the batch job.

30112 The **–O** option allows users to place the message in the standard output stream of the batch job.

30113 Historically, the *qmsg* utility is an existing practice in the offerings of one or more implementors  
30114 of an NQS-derived batch system. The utility has been found to be useful enough that it deserves  
30115 to be included in this volume of IEEE Std 1003.1-2001.

**30116 FUTURE DIRECTIONS**

30117 None.

**30118 SEE ALSO**

30119 Chapter 3 (on page 101), *qselect*

**30120 CHANGE HISTORY**

30121 Derived from IEEE Std 1003.2d-1994.

**30122 Issue 6**

30123 The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

30124 **NAME**

30125       qrerun — rerun batch jobs

30126 **SYNOPSIS**

30127 BE     qrerun *job\_identifier* ...

30128

30129 **DESCRIPTION**

30130       To rerun a batch job is to terminate the session leader of the batch job, delete any associated  
30131       checkpoint files, and return the batch job to the batch queued state. A batch job is rerun by a  
30132       request to the batch server that manages the batch job. The *qrerun* utility is a user-accessible  
30133       batch client that requests the rerunning of one or more batch jobs.

30134       The *qrerun* utility shall rerun those batch jobs for which a batch *job\_identifier* is presented to the  
30135       utility.

30136       The *qrerun* utility shall rerun batch jobs in the order in which their batch *job\_identifiers* are  
30137       presented to the utility.

30138       If the *qrerun* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed  
30139       to process the remaining batch *job\_identifiers*, if any.

30140       The *qrerun* utility shall rerun batch jobs by sending a *Rerun Job Request* to the batch server that  
30141       manages each batch job.

30142       For each successfully processed batch *job\_identifier*, the *qrerun* utility shall have rerun the  
30143       corresponding batch job at the time the utility exits.

30144 **OPTIONS**

30145       None.

30146 **OPERANDS**

30147       The *qrerun* utility shall accept one or more operands that conform to the syntax for a batch  
30148       *job\_identifier* (see Section 3.3.1 (on page 122)).

30149 **STDIN**

30150       Not used.

30151 **INPUT FILES**

30152       None.

30153 **ENVIRONMENT VARIABLES**

30154       The following environment variables shall affect the execution of *qrerun*:

30155       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
30156                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
30157                  Internationalization Variables for the precedence of internationalization variables  
30158                  used to determine the values of locale categories.)

30159       **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
30160                  internationalization variables.

30161       **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
30162                  characters (for example, single-byte as opposed to multi-byte characters in  
30163                  arguments).

30164       **LC\_MESSAGES**

30165                  Determine the locale that should be used to affect the format and contents of  
30166                  diagnostic messages written to standard error.

30167        **LOGNAME** Determine the login name of the user.

**30168 ASYNCHRONOUS EVENTS**

30169        Default.

**30170 STDOUT**

30171        None.

**30172 STDERR**

30173        The standard error shall be used only for diagnostic messages.

**30174 OUTPUT FILES**

30175        None.

**30176 EXTENDED DESCRIPTION**

30177        None.

**30178 EXIT STATUS**

30179        The following exit values shall be returned:

30180        0 Successful completion.

30181        >0 An error occurred.

**30182 CONSEQUENCES OF ERRORS**

30183        In addition to the default behavior, the *qrerun* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qrerun* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-defined.

**30188 APPLICATION USAGE**

30189        None.

**30190 EXAMPLES**

30191        None.

**30192 RATIONALE**

30193        The *qrerun* utility allows users to cause jobs in the running state to exit and rerun.

30194        The *qrerun* utility is a new utility, *vis-a-vis* existing practice, that has been defined in this volume of IEEE Std 1003.1-2001 to correct user-perceived deficiencies in the existing practice.

**30196 FUTURE DIRECTIONS**

30197        None.

**30198 SEE ALSO**

30199        Chapter 3 (on page 101)

**30200 CHANGE HISTORY**

30201        Derived from IEEE Std 1003.2d-1994.

**30202 Issue 6**

30203        The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

## 30204 NAME

30205        qrsls — release batch jobs

## 30206 SYNOPSIS

30207 BE        qrsls [-h hold\_list] job\_identifier ...

## 30209 DESCRIPTION

30210        A batch job might have one or more holds, which prevent the batch job from executing. A batch  
30211        job from which all the holds have been removed becomes eligible for execution and is said to  
30212        have been released. A batch job hold is removed by sending a request to the batch server that  
30213        manages the batch job. The *qrsls* utility is a user-accessible client of batch services that requests  
30214        holds be removed from one or more batch jobs.

30215        The *qrsls* utility shall remove one or more holds from those batch jobs for which a batch  
30216        *job\_identifier* is presented to the utility.

30217        The *qrsls* utility shall remove holds from batch jobs in the order in which their batch *job\_identifiers*  
30218        are presented to the utility.

30219        If the *qrsls* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
30220        process the remaining batch *job\_identifiers*, if any.

30221        The *qrsls* utility shall remove holds on each batch job by sending a *Release Job Request* to the batch  
30222        server that manages the batch job.

30223        The *qrsls* utility shall not exit until the holds have been removed from the batch job  
30224        corresponding to each successfully processed batch *job\_identifier*.

## 30225 OPTIONS

30226        The *qrsls* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
30227        12.2, Utility Syntax Guidelines.

30228        The following option shall be supported by the implementation:

30229        **-h hold\_list** Define the types of holds to be removed from the batch job.

30230        The *qrsls -h* option shall accept a value for the *hold\_list* option-argument that is a  
30231        string of alphanumeric characters in the portable character set (see the Base  
30232        Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

30233        The *qrsls* utility shall accept a value for the *hold\_list* option-argument that is a string  
30234        of one or more of the characters 'u', 's', or 'o', or the single character 'n'.

30235        For each unique character in the *hold\_list* option-argument, the *qrsls* utility shall add  
30236        a value to the *Hold\_Types* attribute of the batch job as follows, each representing a  
30237        different hold type:

30238            u    USER

30239            s    SYSTEM

30240            o    OPERATOR

30241        If any of these characters are duplicated in the *hold\_list* option-argument, the  
30242        duplicates shall be ignored.

30243        An existing *Hold\_Types* attribute can be cleared by the following hold type:

30244            n    NO\_HOLD

30245        The *qrsls* utility shall consider it an error if any hold type other than 'n' is  
30246        combined with hold type 'n'.

30247        Strictly conforming applications shall not repeat any of the characters 'u', 's',  
30248        'o', or 'n' within the *hold\_list* option-argument. The *qrsls* utility shall permit the  
30249        repetition of characters, but shall not assign additional meaning to the repeated  
30250        characters.

30251        An implementation may define other hold types. The conformance document for  
30252        an implementation shall describe any additional hold types, how they are  
30253        specified, their internal behavior, and how they affect the behavior of the utility.

30254        If the **-h** option is not presented to the *qrsls* utility, the implementation shall remove  
30255        the USER hold in the *Hold\_Types* attribute.

## 30256 OPERANDS

30257        The *qrsls* utility shall accept one or more operands that conform to the syntax for a batch  
30258        *job\_identifier* (see Section 3.3.1 (on page 122)).

## 30259 STDIN

30260        Not used.

## 30261 INPUT FILES

30262        None.

## 30263 ENVIRONMENT VARIABLES

30264        The following environment variables shall affect the execution of *qrsls*:

30265        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
30266        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
30267        Internationalization Variables for the precedence of internationalization variables  
30268        used to determine the values of locale categories.)

30269        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
30270        internationalization variables.

30271        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
30272        characters (for example, single-byte as opposed to multi-byte characters in  
30273        arguments).

### 30274 *LC\_MESSAGES*

30275        Determine the locale that should be used to affect the format and contents of  
30276        diagnostic messages written to standard error.

30277        *LOGNAME*     Determine the login name of the user.

## 30278 ASYNCHRONOUS EVENTS

30279        Default.

## 30280 STDOUT

30281        None.

## 30282 STDERR

30283        The standard error shall be used only for diagnostic messages.

## 30284 OUTPUT FILES

30285        None.

**30286 EXTENDED DESCRIPTION**

30287 None.

**30288 EXIT STATUS**

30289 The following exit values shall be returned:

30290 0 Successful completion.

30291 >0 An error occurred.

**30292 CONSEQUENCES OF ERRORS**

30293 In addition to the default behavior, the *qrsls* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qrsls* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-defined.

**30298 APPLICATION USAGE**

30299 None.

**30300 EXAMPLES**

30301 None.

**30302 RATIONALE**

30303 The *qrsls* utility allows users, operators, and administrators to remove holds from jobs.

30304 The *qrsls* utility does not support any job selection options or wildcard arguments. Users may acquire a list of jobs selected by attributes using the *qselect* utility. For example, a user could select all of their held jobs.

30307 The **-h** option allows the user to specify the type of hold that is to be removed. This option allows for USER, SYSTEM, OPERATOR, and implementation-defined hold types. The batch server that manages the batch job will verify whether the user is authorized to remove the specified hold for the batch job. If more than one type of hold has been placed on the batch job, a user may wish to remove only some of them.

30312 Mail is not required on release because the administrator has the tools and libraries to build this option if required.

30314 The *qrsls* utility is a new utility *vis-a-vis* existing practice; it has been defined in this volume of IEEE Std 1003.1-2001 as the natural complement to the *qhold* utility.

**30316 FUTURE DIRECTIONS**

30317 None.

**30318 SEE ALSO**

30319 Chapter 3 (on page 101), *qhold*, *qselect*

**30320 CHANGE HISTORY**

30321 Derived from IEEE Std 1003.2d-1994.

**30322 Issue 6**

30323 The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

## 30324 NAME

30325 qselect — select batch jobs

## 30326 SYNOPSIS

30327 BE        qselect [-a [op]date\_time][-A account\_string][-c [op]interval]  
 30328            [-h hold\_list][-l resource\_list][-N name][-p [op]priority]  
 30329            [-q destination][-r y|n][-s states][-u user\_list]

30330

## 30331 DESCRIPTION

30332 To select a set of batch jobs is to return the batch *job\_identifiers* for each batch job that meets a list  
 30333 of selection criteria. A set of batch jobs is selected by a request to a batch server. The *qselect*  
 30334 utility is a user-accessible batch client that requests the selection of batch jobs.

30335 Upon successful completion, the *qselect* utility shall have returned a list of zero or more batch  
 30336 *job\_identifiers* that meet the criteria specified by the options and option-arguments presented to  
 30337 the utility.

30338 The *qselect* utility shall select batch jobs by sending a *Select Jobs Request* to a batch server. The  
 30339 *qselect* utility shall not exit until the server replies to each request generated.

30340 For each option presented to the *qselect* utility, the utility shall restrict the set of selected batch  
 30341 jobs as described in the OPTIONS section.

30342 The *qselect* utility shall not restrict selection of batch jobs except by authorization and as required  
 30343 by the options presented to the utility.

30344 When an option is specified with a mandatory or optional *op* component to the option-  
 30345 argument, then *op* shall specify a relation between the value of a certain batch job attribute and the  
 30346 *value* component of the option-argument. If an *op* is allowable on an option, then the  
 30347 description of the option letter indicates the *op* as either mandatory or optional. Acceptable  
 30348 strings for the *op* component, and the relation the string indicates, are shown in the following  
 30349 list:

- 30350 .eq. The value represented by the attribute of the batch job is equal to the value represented  
 30351 by the option-argument.
- 30352 .ge. The value represented by the attribute of the batch job is greater than or equal to the  
 30353 value represented by the option-argument.
- 30354 .gt. The value represented by the attribute of the batch job is greater than the value  
 30355 represented by the option-argument.
- 30356 .lt. The value represented by the attribute of the batch job is less than the value  
 30357 represented by the option-argument.
- 30358 .le. The value represented by the attribute of the batch job is less than or equal to the value  
 30359 represented by the option-argument.
- 30360 .ne. The value represented by the attribute of the batch job is not equal to the value  
 30361 represented by the option-argument.

## 30362 OPTIONS

30363 The *qselect* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 30364 12.2, Utility Syntax Guidelines.

30365 The following options shall be supported by the implementation:

30366 **-a [op]date\_time**  
 30367                      Restrict selection to a specific time, or a range of times.

30368        The *qselect* utility shall select only batch jobs for which the value of the  
30369        *Execution\_Time* attribute is related to the Epoch equivalent of the local time  
30370        expressed by the value of the *date\_time* component of the option-argument in the  
30371        manner indicated by the value of the *op* component of the option-argument.

30372        The *qselect* utility shall accept a *date\_time* component of the option-argument that  
30373        conforms to the syntax of the *time* operand of the *touch* utility.

30374        If the *op* component of the option-argument is not presented to the *qselect* utility,  
30375        the utility shall select batch jobs for which the *Execution\_Time* attribute is equal to  
30376        the *date\_time* component of the option-argument.

30377        When comparing times, the *qselect* utility shall use the following definitions for the  
30378        *op* component of the option-argument:

- 30379        .eq.      The time represented by value of the *Execution\_Time* attribute of the batch  
30380        job is equal to the time represented by the *date\_time* component of the  
30381        option-argument.
- 30382        .ge.      The time represented by value of the *Execution\_Time* attribute of the batch  
30383        job is after or equal to the time represented by the *date\_time* component of  
30384        the option-argument.
- 30385        .gt.      The time represented by value of the *Execution\_Time* attribute of the batch  
30386        job is after the time represented by the *date\_time* component of the  
30387        option-argument.
- 30388        .lt.      The time represented by value of the *Execution\_Time* attribute of the batch  
30389        job is before the time represented by the *date\_time* component of the  
30390        option-argument.
- 30391        .le.      The time represented by value of the *Execution\_Time* attribute of the batch  
30392        job is before or equal to the time represented by the *date\_time* component of  
30393        the option-argument.
- 30394        .ne.      The time represented by value of the *Execution\_Time* attribute of the batch  
30395        job is not equal to the time represented by the *date\_time* component of the  
30396        option-argument.

30397        The *qselect* utility shall accept the defined character strings for the *op* component of  
30398        the option-argument.

30399        **-A account\_string**

30400        Restrict selection to the batch jobs charging a specified account.

30401        The *qselect* utility shall select only batch jobs for which the value of the  
30402        *Account\_Name* attribute of the batch job matches the value of the *account\_string*  
30403        option-argument.

30404        The syntax of the *account\_string* option-argument is unspecified.

30405        **-c [op]interval**

30406        Restrict selection to batch jobs within a range of checkpoint intervals.

30407        The *qselect* utility shall select only batch jobs for which the value of the *CHECKPOINT*  
30408        attribute relates to the value of the *interval* component of the option-argument in  
30409        the manner indicated by the value of the *op* component of the option-argument.

30410        If the *op* component of the option-argument is omitted, the *qselect* utility shall  
30411        select batch jobs for which the value of the *CHECKPOINT* attribute is equal to the value

- 30412 of the *interval* component of the option-argument.
- 30413 When comparing checkpoint intervals, the *qselect* utility shall use the following  
30414 definitions for the *op* component of the option-argument:
- 30415 .eq. The value of the *Clockpoint* attribute of the batch job equals the value of  
30416 the *interval* component of the option-argument.
- 30417 .ge. The value of the *Clockpoint* attribute of the batch job is greater than or  
30418 equal to the value of the *interval* component option-argument.
- 30419 .gt. The value of the *Clockpoint* attribute of the batch job is greater than the  
30420 value of the *interval* component option-argument.
- 30421 .lt. The value of the *Clockpoint* attribute of the batch job is less than the value  
30422 of the *interval* component option-argument.
- 30423 .le. The value of the *Clockpoint* attribute of the batch job is less than or equal  
30424 to the value of the *interval* component option-argument.
- 30425 .ne. The value of the *Clockpoint* attribute of the batch job does not equal the  
30426 value of the *interval* component option-argument.
- 30427 The *qselect* utility shall accept the defined character strings for the *op* component of  
30428 the option-argument.
- 30429 The ordering relationship for the values of the interval option-argument is defined  
30430 to be:
- 30431 'n' .gt. 's' .gt. 'c=minutes' .ge. 'c'
- 30432 When comparing *Clockpoint* attributes with an interval having the value of the  
30433 single character 'u', only equality or inequality are valid comparisons.
- 30434 **-h hold\_list** Restrict selection to batch jobs that have a specific type of hold.
- 30435 The *qselect* utility shall select only batch jobs for which the value of the *Hold\_Types*  
30436 attribute matches the value of the *hold\_list* option-argument.
- 30437 The *qselect* -h option shall accept a value for the *hold\_list* option-argument that is a  
30438 string of alphanumeric characters in the portable character set (see the Base  
30439 Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).
- 30440 The *qselect* utility shall accept a value for the *hold\_list* option-argument that is a  
30441 string of one or more of the characters 'u', 's', or 'o', or the single character  
30442 'n'.
- 30443 Each unique character in the *hold\_list* option-argument of the *qselect* utility is  
30444 defined as follows, each representing a different hold type:
- 30445 u USER
- 30446 s SYSTEM
- 30447 o OPERATOR
- 30448 If any of these characters are duplicated in the *hold\_list* option-argument, the  
30449 duplicates shall be ignored.
- 30450 The *qselect* utility shall consider it an error if any hold type other than 'n' is  
30451 combined with hold type 'n'.

30452 Strictly conforming applications shall not repeat any of the characters 'u', 's',  
30453 'o', or 'n' within the *hold\_list* option-argument. The *qselect* utility shall permit  
30454 the repetition of characters, but shall not assign additional meaning to the repeated  
30455 characters.

30456 An implementation may define other hold types. The conformance document for  
30457 an implementation shall describe any additional hold types, how they are  
30458 specified, their internal behavior, and how they affect the behavior of the utility.

30459 **-l resource\_list**

30460 Restrict selection to batch jobs with specified resource limits and attributes.

30461 The *qselect* utility shall accept a *resource\_list* option-argument with the following  
30462 syntax:

30463 *resource\_name op value [,,resource\_name op value,, ...]*

30464 When comparing resource values, the *qselect* utility shall use the following  
30465 definitions for the *op* component of the option-argument:

- 30466 .eq. The value of the resource of the same name in the *Resource\_List* attribute  
30467 of the batch job equals the value of the *value* component of the option-  
30468 argument.
- 30469 .ge. The value of the resource of the same name in the *Resource\_List* attribute  
30470 of the batch job is greater than or equal to the value of the *value*  
30471 component of the option-argument.
- 30472 .gt. The value of the resource of the same name in the *Resource\_List* attribute  
30473 of the batch job is greater than the value of the *value* component of the  
30474 option-argument.
- 30475 .lt. The value of the resource of the same name in the *Resource\_List* attribute  
30476 of the batch job is less than the value of the *value* component of the  
30477 option-argument.
- 30478 .ne. The value of the resource of the same name in the *Resource\_List* attribute  
30479 of the batch job does not equal the value of the *value* component of the  
30480 option-argument.
- 30481 .le. The value of the resource of the same name in the *Resource\_List* attribute  
30482 of the batch job is less than or equal to the value of the *value* component  
30483 of the option-argument.

30484 When comparing the limit of a *Resource\_List* attribute with the *value* component of  
30485 the option-argument, if the limit, the value, or both are non-numeric, only equality  
30486 or inequality are valid comparisons.

30487 The *qselect* utility shall select only batch jobs for which the values of the  
30488 *resource\_names* listed in the *resource\_list* option-argument match the corresponding  
30489 limits of the *Resource\_List* attribute of the batch job.

30490 Limits of *resource\_names* present in the *Resource\_List* attribute of the batch job that  
30491 have no corresponding values in the *resource\_list* option-argument shall not be  
30492 considered when selecting batch jobs.

30493 **-N name** Restrict selection to batch jobs with a specified name.

30494 The *qselect* utility shall select only batch jobs for which the value of the *Job\_Name*  
30495 attribute matches the value of the *name* option-argument. The string specified in

30496           the *name* option-argument shall be passed, uninterpreted, to the server. This allows  
 30497           an implementation to match “wildcard” patterns against batch job names.

30498           An implementation shall describe in the conformance document the format it  
 30499           supports for matching against the *Job\_Name* attribute.

30500           **-p [op]priority**

30501           Restrict selection to batch jobs of the specified priority or range of priorities.

30502           The *qselect* utility shall select only batch jobs for which the value of the *Priority*  
 30503           attribute of the batch job relates to the value of the *priority* component of the  
 30504           option-argument in the manner indicated by the value of the *op* component of the  
 30505           option-argument.

30506           If the *op* component of the option-argument is omitted, the *qselect* utility shall  
 30507           select batch jobs for which the value of the *Priority* attribute of the batch job is  
 30508           equal to the value of the *priority* component of the option-argument.

30509           When comparing priority values, the *qselect* utility shall use the following  
 30510           definitions for the *op* component of the option-argument:

- 30511           . eq.     The value of the *Priority* attribute of the batch job equals the value of the  
                   30512           *priority* component of the option-argument.
- 30513           . ge.     The value of the *Priority* attribute of the batch job is greater than or equal  
                   30514           to the value of the *priority* component option-argument.
- 30515           . gt.     The value of the *Priority* attribute of the batch job is greater than the value  
                   30516           of the *priority* component option-argument.
- 30517           . lt.     The value of the *Priority* attribute of the batch job is less than the value of  
                   30518           the *priority* component option-argument.
- 30519           . lt.     The value of the *Priority* attribute of the batch job is less than or equal to  
                   30520           the value of the *priority* component option-argument.
- 30521           . ne.     The value of the *Priority* attribute of the batch job does not equal the value  
                   30522           of the *priority* component option-argument.

30523           **-q destination**

30524           Restrict selection to the specified batch queue or server, or both.

30525           The *qselect* utility shall select only batch jobs that are located at the destination  
 30526           indicated by the value of the *destination* option-argument.

30527           The destination defines a batch queue, a server, or a batch queue at a server.

30528           The *qselect* utility shall accept an option-argument for the **-q** option that conforms  
 30529           to the syntax for a destination. If the **-q** option is not presented to the *qselect* utility,  
 30530           the utility shall select batch jobs from all batch queues at the default batch server.

30531           If the option-argument describes only a batch queue, the *qselect* utility shall select  
 30532           only batch jobs from the batch queue of the specified name at the default batch  
 30533           server. The means by which *qselect* determines the default server is  
 30534           implementation-defined.

30535           If the option-argument describes only a batch server, the *qselect* utility shall select  
 30536           batch jobs from all the batch queues at that batch server.

30537           If the option-argument describes both a batch queue and a batch server, the *qselect*  
 30538           utility shall select only batch jobs from the specified batch queue at the specified

|       |                                                                                                                                                                                                                                                      |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30539 | server.                                                                                                                                                                                                                                              |
| 30540 | <b>-r Y   n</b> Restrict selection to batch jobs with the specified rerunability status.                                                                                                                                                             |
| 30541 | The <i>qselect</i> utility shall select only batch jobs for which the value of the <i>Rerunable</i> attribute of the batch job matches the value of the option-argument.                                                                             |
| 30542 |                                                                                                                                                                                                                                                      |
| 30543 | The <i>qselect</i> utility shall accept a value for the option-argument that consists of either the single character 'Y' or the single character 'n'. The character 'Y' represents the value TRUE, and the character 'n' represents the value FALSE. |
| 30544 |                                                                                                                                                                                                                                                      |
| 30545 |                                                                                                                                                                                                                                                      |
| 30546 | <b>-s states</b> Restrict selection to batch jobs in the specified states.                                                                                                                                                                           |
| 30547 | The <i>qselect</i> utility shall accept an option-argument that consists of any combination of the characters 'e', 'q', 'r', 'w', 'h', and 't'.                                                                                                      |
| 30548 |                                                                                                                                                                                                                                                      |
| 30549 | Conforming applications shall not repeat any character in the option-argument.                                                                                                                                                                       |
| 30550 | The <i>qselect</i> utility shall permit the repetition of characters in the option-argument, but shall not assign additional meaning to repeated characters.                                                                                         |
| 30551 |                                                                                                                                                                                                                                                      |
| 30552 | The <i>qselect</i> utility shall interpret the characters in the <i>states</i> option-argument as follows:                                                                                                                                           |
| 30553 |                                                                                                                                                                                                                                                      |
| 30554 | e    Represents the EXITING state.                                                                                                                                                                                                                   |
| 30555 | q    Represents the QUEUED state.                                                                                                                                                                                                                    |
| 30556 | r    Represents the RUNNING state.                                                                                                                                                                                                                   |
| 30557 | t    Represents the TRANSITING state.                                                                                                                                                                                                                |
| 30558 | h    Represents the HELD state.                                                                                                                                                                                                                      |
| 30559 | w    Represents the WAITING state.                                                                                                                                                                                                                   |
| 30560 | For each character in the <i>states</i> option-argument, the <i>qselect</i> utility shall select batch jobs in the corresponding state.                                                                                                              |
| 30561 |                                                                                                                                                                                                                                                      |
| 30562 | <b>-u user_list</b> Restrict selection to batch jobs owned by the specified user names.                                                                                                                                                              |
| 30563 | The <i>qselect</i> utility shall select only the batch jobs of those users specified in the <i>user_list</i> option-argument.                                                                                                                        |
| 30564 |                                                                                                                                                                                                                                                      |
| 30565 | The <i>qselect</i> utility shall accept a <i>user_list</i> option-argument that conforms to the following syntax:                                                                                                                                    |
| 30566 |                                                                                                                                                                                                                                                      |
| 30567 | <i>username[@host][, , username[@host], , . . . ]</i>                                                                                                                                                                                                |
| 30568 | The <i>qselect</i> utility shall accept only one user name that is missing a corresponding host name. The <i>qselect</i> utility shall accept only one user name per named host.                                                                     |
| 30569 |                                                                                                                                                                                                                                                      |
| 30570 | <b>OPERANDS</b>                                                                                                                                                                                                                                      |
| 30571 | None.                                                                                                                                                                                                                                                |
| 30572 | <b>STDIN</b>                                                                                                                                                                                                                                         |
| 30573 | Not used.                                                                                                                                                                                                                                            |
| 30574 | <b>INPUT FILES</b>                                                                                                                                                                                                                                   |
| 30575 | None.                                                                                                                                                                                                                                                |

**30576 ENVIRONMENT VARIABLES**

- 30577       The following environment variables shall affect the execution of *qselect*:
- 30578       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
30579                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
30580                  Internationalization Variables for the precedence of internationalization variables  
30581                  used to determine the values of locale categories.)
- 30582       **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
30583                  internationalization variables.
- 30584       **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
30585                  characters (for example, single-byte as opposed to multi-byte characters in  
30586                  arguments).
- 30587       **LC\_MESSAGES**   Determine the locale that should be used to affect the format and contents of  
30588                  diagnostic messages written to standard error.
- 30589       **LOGNAME**     Determine the login name of the user.
- 30590       **TZ**          Determine the timezone used to interpret the *date-time* option-argument. If *TZ* is  
30591                  unset or null, an unspecified default timezone shall be used.

**30593 ASYNCHRONOUS EVENTS**

- 30594       Default.

**30595 STDOUT**

- 30596       The *qselect* utility shall write zero or more batch *job\_identifiers* to standard output.
- 30597       The *qselect* utility shall separate the batch *job\_identifiers* written to standard output by white  
30598                  space.
- 30599       The *qselect* utility shall write batch *job\_identifiers* in the following format:
- 30600                  *sequence\_number.server\_name@server*

**30601 STDERR**

- 30602       The standard error shall be used only for diagnostic messages.

**30603 OUTPUT FILES**

- 30604       None.

**30605 EXTENDED DESCRIPTION**

- 30606       None.

**30607 EXIT STATUS**

- 30608       The following exit values shall be returned:
- 30609                  0   Successful completion.
- 30610                  >0   An error occurred.

**30611 CONSEQUENCES OF ERRORS**

- 30612       Default.

## 30613 APPLICATION USAGE

30614 None.

## 30615 EXAMPLES

30616 The following example shows how a user might use the *qselect* utility in conjunction with the  
30617 *qdel* utility to delete all of his or her jobs in the queued state without affecting any jobs that are  
30618 already running:

30619 `qdel $(qselect -s q)`

30620 or:

30621 `qselect -s q || xargs qdel`

## 30622 RATIONALE

30623 The *qselect* utility allows users to acquire a list of job identifiers that match user-specified  
30624 selection criteria. The list of identifiers returned by the *qselect* utility conforms to the syntax of  
30625 the batch job identifier list processed by a utility such as *qmove*, *qdel*, and *qrsl*. The *qselect* utility is  
30626 thus a powerful tool for causing another batch system utility to act upon a set of jobs that match  
30627 a list of selection criteria.

30628 The options of the *qselect* utility let the user apply a number of useful filters for selecting jobs.  
30629 Each option further restricts the selection of jobs. Many of the selection options allow the  
30630 specification of a relational operator. The FORTRAN-like syntax of the operator—that is,  
30631 ".lt."—was chosen rather than the C-like "<=" meta-characters.

30632 The **-a** option allows users to restrict the selected jobs to those that have been submitted (or  
30633 altered) to wait until a particular time. The time period is determined by the argument of this  
30634 option, which includes both a time and an operator—it is thus possible to select jobs waiting  
30635 until a specific time, jobs waiting until after a certain time, or those waiting for a time before the  
30636 specified time.

30637 The **-A** option allows users to restrict the selected jobs to those that have been submitted (or  
30638 altered) to charge a particular account.

30639 The **-c** option allows users to restrict the selected jobs to those whose checkpointing interval  
30640 falls within the specified range.

30641 The **-l** option allows users to select those jobs whose resource limits fall within the range  
30642 indicated by the value of the option. For example, a user could select those jobs for which the  
30643 CPU time limit is greater than two hours.

30644 The **-N** option allows users to select jobs by job name. For instance, all the parts of a task that  
30645 have been divided in parallel jobs might be given the same name, and thus manipulated as a  
30646 group by means of this option.

30647 The **-q** option allows users to select jobs in a specified queue.

30648 The **-r** option allows users to select only those jobs with a specified rerun criteria. For instance, a  
30649 user might select only those jobs that can be rerun for use with the *qrerun* utility.

30650 The **-s** option allows users to select only those jobs that are in a certain state.

30651 The **-u** option allows users to select jobs that have been submitted to execute under a particular  
30652 account.

30653 The selection criteria provided by the options of the *qselect* utility allow users to select jobs based  
30654 on all the appropriate attributes that can be assigned to jobs by the *qsub* utility.

30655 Historically, the *qselect* utility has not been a part of existing practice; it is an improvement that  
30656 has been introduced in this volume of IEEE Std 1003.1-2001.

**30657 FUTURE DIRECTIONS**

30658        None.

**30659 SEE ALSO**

30660        *qdel, qrerun, qrls, qselect, qsub, touch*, Chapter 3 (on page 101)

**30661 CHANGE HISTORY**

30662        Derived from IEEE Std 1003.2d-1994.

30663 **NAME**

30664        *qsig* — signal batch jobs

30665 **SYNOPSIS**

30666 BE        *qsig [-s signal] job\_identifier ...*

30667

30668 **DESCRIPTION**

30669        To signal a batch job is to send a signal to the session leader of the batch job. A batch job is  
30670        signaled by sending a request to the batch server that manages the batch job. The *qsig* utility is a  
30671        user-accessible batch client that requests the signaling of a batch job.

30672        The *qsig* utility shall signal those batch jobs for which a batch *job\_identifier* is presented to the  
30673        utility. The *qsig* utility shall not signal any batch jobs whose batch *job\_identifiers* are not  
30674        presented to the utility.

30675        The *qsig* utility shall signal batch jobs in the order in which the corresponding batch  
30676        *job\_identifiers* are presented to the utility. If the *qsig* utility fails to process a batch *job\_identifier*  
30677        successfully, the utility shall proceed to process the remaining batch *job\_identifiers*, if any.

30678        The *qsig* utility shall signal batch jobs by sending a *Signal Job Request* to the batch server that  
30679        manages the batch job.

30680        For each successfully processed batch *job\_identifier*, the *qsig* utility shall have received a  
30681        completion reply to each *Signal Job Request* sent to a batch server at the time the utility exits.

30682 **OPTIONS**

30683        The *qsig* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
30684        12.2, Utility Syntax Guidelines.

30685        The following option shall be supported by the implementation:

30686        **-s signal**        Define the signal to be sent to the batch job.

30687        The *qsig* utility shall accept a *signal* option-argument that is either a symbolic  
30688        signal name or an unsigned integer signal number (see the POSIX.1-1990 standard,  
30689        Section 3.3.1.1). The *qsig* utility shall accept signal names for which the SIG prefix  
30690        has been omitted.

30691        If the *signal* option-argument is a signal name, the *qsig* utility shall send that name.

30692        If the *signal* option-argument is a number, the *qsig* utility shall send the signal  
30693        value represented by the number.

30694        If the **-s** option is not presented to the *qsig* utility, the utility shall send the signal  
30695        SIGTERM to each signaled batch job.

30696 **OPERANDS**

30697        The *qsig* utility shall accept one or more operands that conform to the syntax for a batch  
30698        *job\_identifier* (see Section 3.3.1 (on page 122)).

30699 **STDIN**

30700        Not used.

30701 **INPUT FILES**

30702        None.

**30703 ENVIRONMENT VARIABLES**

30704 The following environment variables shall affect the execution of *qsig*:

30705 **LANG** Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

30709 **LC\_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

30711 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

**30714 LC\_MESSAGES**

30715 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

30717 **LOGNAME** Determine the login name of the user.

**30718 ASYNCHRONOUS EVENTS**

30719 Default.

**30720 STDOUT**

30721 An implementation of the *qsig* utility may write informative messages to standard output.

**30722 STDERR**

30723 The standard error shall be used only for diagnostic messages.

**30724 OUTPUT FILES**

30725 None.

**30726 EXTENDED DESCRIPTION**

30727 None.

**30728 EXIT STATUS**

30729 The following exit values shall be returned:

30730 0 Successful completion.

30731 >0 An error occurred.

**30732 CONSEQUENCES OF ERRORS**

30733 In addition to the default behavior, the *qsig* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qsig* utility waits to output the diagnostic message while attempting to locate the batch job on other servers is implementation-defined.

**30738 APPLICATION USAGE**

30739 None.

**30740 EXAMPLES**

30741 None.

**30742 RATIONALE**

30743 The *qsig* utility allows users to signal batch jobs.

30744 A user may be unable to signal a batch job with the *kill* utility of the operating system for a number of reasons. First, the process ID of the batch job may be unknown to the user. Second,

30746 the processes of the batch job may be on a remote node. However, by virtue of communication  
30747 between batch nodes, the *qsig* utility can arrange for the signaling of a process.

30748 Because a batch job that is not running cannot be signaled, and because the signal may not  
30749 terminate the batch job, the *qsig* utility is not a substitute for the *qdel* utility.

30750 The options of the *qsig* utility allow the user to specify the signal that is to be sent to the batch  
30751 job.

30752 The **-s** option allows users to specify a signal by name or by number, and thus override the  
30753 default signal. The POSIX.1-1990 standard defines signals by both name and number.

30754 The *qsig* utility is a new utility, *vis-a-vis* existing practice; it has been defined in this volume of  
30755 IEEE Std 1003.1-2001 in response to user-perceived shortcomings in existing practice.

30756 **FUTURE DIRECTIONS**

30757 None.

30758 **SEE ALSO**

30759 Chapter 3 (on page 101), *kill*, *qdel*

30760 **CHANGE HISTORY**

30761 Derived from IEEE Std 1003.2d-1994.

30762 **Issue 6**

30763 The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.

**30764 NAME**

30765 qstat — show status of batch jobs

**30766 SYNOPSIS**

30767 BE qstat [-f] *job\_identifier* ...

30768 qstat -Q [-f] *destination* ...

30769 qstat -B [-f] *server\_name* ...

30770

**30771 DESCRIPTION**

30772 The status of a batch job, batch queue, or batch server is obtained by a request to the server. The  
30773 *qstat* utility is a user-accessible batch client that requests the status of one or more batch jobs,  
30774 batch queues, or servers, and writes the status information to standard output.

30775 For each successfully processed batch *job\_identifier*, the *qstat* utility shall display information  
30776 about the corresponding batch job.

30777 For each successfully processed destination, the *qstat* utility shall display information about the  
30778 corresponding batch queue.

30779 For each successfully processed server name, the *qstat* utility shall display information about the  
30780 corresponding server.

30781 The *qstat* utility shall acquire batch job status information by sending a *Job Status Request* to a  
30782 batch server. The *qstat* utility shall acquire batch queue status information by sending a *Queue*  
30783 *Status Request* to a batch server. The *qstat* utility shall acquire server status information by  
30784 sending a *Server Status Request* to a batch server.

**30785 OPTIONS**

30786 The *qstat* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
30787 12.2, Utility Syntax Guidelines.

30788 The following options shall be supported by the implementation:

30789 **-f** Specify that a full display is produced.

30790 The minimum contents of a full display are specified in the STDOUT section.

30791 Additional contents and format of a full display are implementation-defined.

30792 **-Q** Specify that the operand is a destination.

30793 The *qstat* utility shall display information about each batch queue at each  
30794 destination identified as an operand.

30795 **-B** Specify that the operand is a server name.

30796 The *qstat* utility shall display information about each server identified as an  
30797 operand.

**30798 OPERANDS**

30799 If the **-Q** option is presented to the *qstat* utility, the utility shall accept one or more operands that  
30800 conform to the syntax for a destination (see Section 3.3.2 (on page 123)).

30801 If the **-B** option is presented to the *qstat* utility, the utility shall accept one or more *server\_name*  
30802 operands.

30803 If neither the **-B** nor the **-Q** option is presented to the *qstat* utility, the utility shall accept one or  
30804 more operands that conform to the syntax for a batch *job\_identifier* (see Section 3.3.1 (on page  
30805 122)).

30806 **STDIN**

30807 Not used.

30808 **INPUT FILES**

30809 None.

30810 **ENVIRONMENT VARIABLES**

30811 The following environment variables shall affect the execution of *qstat*:

30812 *HOME* Determine the pathname of the user's home directory.

30813 *LANG* Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

30817 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
30818 internationalization variables.

30819 *LC\_COLLATE*

30820 Determine the locale for the behavior of ranges, equivalence classes, and multi-  
30821 character collating elements within regular expressions.

30822 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
30823 characters (for example, single-byte as opposed to multi-byte characters in  
30824 arguments).

30825 *LC\_MESSAGES*

30826 Determine the locale that should be used to affect the format and contents of  
30827 diagnostic messages written to standard error.

30828 *LC\_NUMERIC*

30829 Determine the locale for selecting the radix character used when writing floating-  
30830 point formatted output.

30831 **ASYNCHRONOUS EVENTS**

30832 Default.

30833 **STDOUT**

30834 If an operand presented to the *qstat* utility is a batch *job\_identifier* and the **-f** option is not  
30835 specified, the *qstat* utility shall display the following items on a single line, in the stated order,  
30836 with white space between each item, for each successfully processed operand:

- 30837 • The batch *job\_identifier*
- 30838 • The batch job name
- 30839 • The *Job\_Owner* attribute
- 30840 • The CPU time used by the batch job
- 30841 • The batch job state
- 30842 • The batch job location

30843 If an operand presented to the *qstat* utility is a batch *job\_identifier* and the **-f** option is specified,  
30844 the *qstat* utility shall display the following items for each success fully processed operand:

- 30845 • The batch *job\_identifier*
- 30846 • The batch job name

- 30847     • The *Job\_Owner* attribute
  - 30848     • The execution user ID
  - 30849     • The CPU time used by the batch job
  - 30850     • The batch job state
  - 30851     • The batch job location
  - 30852     • Additional implementation-defined information, if any, about the batch job or batch queue
- 30853 If an operand presented to the *qstat* utility is a destination, the **-Q** option is specified, and the **-f** option is not specified, the *qstat* utility shall display the following items on a single line, in the stated order, with white space between each item, for each successfully processed operand:
- 30856     • The batch queue name
  - 30857     • The maximum number of batch jobs that shall be run in the batch queue concurrently
  - 30858     • The total number of batch jobs in the batch queue
  - 30859     • The status of the batch queue
  - 30860     • For each state, the number of batch jobs in that state in the batch queue and the name of the state
  - 30862     • The type of batch queue (execution or routing)
- 30863 If the operands presented to the *qstat* utility are destinations, the **-Q** option is specified, and the **-f** option is specified, the *qstat* utility shall display the following items for each successfully processed operand:
- 30866     • The batch queue name
  - 30867     • The maximum number of batch jobs that shall be run in the batch queue concurrently
  - 30868     • The total number of batch jobs in the batch queue
  - 30869     • The status of the batch queue
  - 30870     • For each state, the number of batch jobs in that state in the batch queue and the name of the state
  - 30872     • The type of batch queue (execution or routing)
  - 30873     • Additional implementation-defined information, if any, about the batch queue
- 30874 If the operands presented to the *qstat* utility are batch server names, the **-B** option is specified, and the **-f** option is not specified, the *qstat* utility shall display the following items on a single line, in the stated order, with white space between each item, for each successfully processed operand:
- 30878     • The batch server name
  - 30879     • The maximum number of batch jobs that shall be run in the batch queue concurrently
  - 30880     • The total number of batch jobs managed by the batch server
  - 30881     • The status of the batch server
  - 30882     • For each state, the number of batch jobs in that state and the name of the state
- 30883 If the operands presented to the *qstat* utility are server names, the **-B** option is specified, and the **-f** option is specified, the *qstat* utility shall display the following items for each successfully processed operand:

- 30886     • The server name
- 30887     • The maximum number of batch jobs that shall be run in the batch queue concurrently
- 30888     • The total number of batch jobs managed by the server
- 30889     • The status of the server
- 30890     • For each state, the number of batch jobs in that state and the name of the state
- 30891     • Additional implementation-defined information, if any, about the server

**30892 STDERR**

30893     The standard error shall be used only for diagnostic messages.

**30894 OUTPUT FILES**

30895     None.

**30896 EXTENDED DESCRIPTION**

30897     None.

**30898 EXIT STATUS**

30899     The following exit values shall be returned:

30900       0   Successful completion.

30901       >0   An error occurred.

**30902 CONSEQUENCES OF ERRORS**

30903     In addition to the default behavior, the *qstat* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qstat* utility waits to output the diagnostic message while attempting to locate the batch job on other servers is implementation-defined.

**30908 APPLICATION USAGE**

30909     None.

**30910 EXAMPLES**

30911     None.

**30912 RATIONALE**

30913     The *qstat* utility allows users to display the status of jobs and list the batch jobs in queues.

30914     The operands of the *qstat* utility may be either job identifiers, queues (specified as destination identifiers), or batch server names. The **-Q** and **-B** options, or absence thereof, indicate the nature of the operands.

30917     The other options of the *qstat* utility allow the user to control the amount of information displayed and the format in which it is displayed. Should a user wish to display the status of a set of jobs that match a selection criteria, the *qselect* utility may be used to acquire such a list.

30920     The **-f** option allows users to request a “full” display in an implementation-defined format.

30921     Historically, the *qstat* utility has been a part of the NQS and its derivatives, the existing practice on which it is based.

**30923 FUTURE DIRECTIONS**

30924     None.

**30925 SEE ALSO**

30926 Chapter 3 (on page 101), *qselect*

**30927 CHANGE HISTORY**

30928 Derived from IEEE Std 1003.2d-1994.

**30929 Issue 6**

30930 IEEE PASC Interpretation 1003.2 #191 is applied, removing the following ENVIRONMENT  
30931 VARIABLES listed as affecting *qstat*: *COLUMNS*, *LINES*, *LOGNAME*, *TERM*, and *TZ*.

30932 The *LC\_TIME* entry is also removed from the ENVIRONMENT VARIABLES section.

**30933 NAME**

30934        qsub — submit a script

**30935 SYNOPSIS**

```
30936 BE qsub [-a date_time][-A account_string][-c interval]
30937 [-C directive_prefix][-e path_name][-h][-j join_list][-k keep_list]
30938 [-m mail_options][-M mail_list][-N name]
30939 [-o path_name][-p priority][-q destination][-r y|n]
30940 [-S path_name_list][-u user_list][-v variable_list][-V]
30941 [-z][script]
```

30942

**30943 DESCRIPTION**

30944        To submit a script is to create a batch job that executes the script. A script is submitted by a  
30945 request to a batch server. The *qsub* utility is a user-accessible batch client that submits a script.

30946        Upon successful completion, the *qsub* utility shall have created a batch job that will execute the  
30947 submitted script.

30948        The *qsub* utility shall submit a script by sending a *Queue Job Request* to a batch server.

30949        The *qsub* utility shall place the value of the following environment variables in the *Variable\_List*  
30950 attribute of the batch job: *HOME*, *LANG*, *LOGNAME*, *PATH*, *MAIL*, *SHELL*, and *TZ*. The name  
30951 of the environment variable shall be the current name prefixed with the string *PBS\_O\_*.

30952        **Note:** If the current value of the *HOME* variable in the environment space of the *qsub* utility is  
30953            /aa/bb/cc, then *qsub* shall place *PBS\_O\_HOME=/aa/bb/cc* in the *Variable\_List* attribute of the  
30954 batch job.

30955        In addition to the variables described above, the *qsub* utility shall add the following variables  
30956 with the indicated values to the variable list:

30957        *PBS\_O\_WORKDIR*      The absolute path of the current working directory of the *qsub* utility  
30958            process.

30959        *PBS\_O\_HOST*          The name of the host on which the *qsub* utility is running.

**30960 OPTIONS**

30961        The *qsub* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
30962 12.2, Utility Syntax Guidelines.

30963        The following options shall be supported by the implementation:

30964        **-a** *date\_time* Define the time at which a batch job becomes eligible for execution.

30965        The *qsub* utility shall accept an option-argument that conforms to the syntax of the  
30966 *time* operand of the *touch* utility.

30967

**Table 4-18** Environment Variable Values (Utilities)

30968

30969

30970

30971

30972

30973

30974

30975

30976

30977

| Variable Name        | Value at qsub Time        |
|----------------------|---------------------------|
| <i>PBS_O_HOME</i>    | <i>HOME</i>               |
| <i>PBS_O_HOST</i>    | Client host name          |
| <i>PBS_O_LANG</i>    | <i>LANG</i>               |
| <i>PBS_O_LOGNAME</i> | <i>LOGNAME</i>            |
| <i>PBS_O_PATH</i>    | <i>PATH</i>               |
| <i>PBS_O_MAIL</i>    | <i>MAIL</i>               |
| <i>PBS_O_SHELL</i>   | <i>SHELL</i>              |
| <i>PBS_O_TZ</i>      | <i>TZ</i>                 |
| <i>PBS_O_WORKDIR</i> | Current working directory |

30978

30979

**Note:** The server that initiates execution of the batch job will add other variables to the batch job's environment; see Section 3.2.2.1 (on page 106).

30980

30981

30982

30983

The *qsub* utility shall set the *Execution\_Time* attribute of the batch job to the number of seconds since the Epoch that is equivalent to the local time expressed by the value of the *date\_time* option-argument. The Epoch is defined in the Base Definitions volume of IEEE Std 1003.1-2001, Section 3.149, Epoch.

30984

30985

30986

If the **-a** option is not presented to the *qsub* utility, the utility shall set the *Execution\_Time* attribute of the batch job to a time (number of seconds since the Epoch) that is earlier than the time at which the utility exits.

30987

**-A account\_string**

30988

30989

Define the account to which the resource consumption of the batch job should be charged.

30990

The syntax of the *account\_string* option-argument is unspecified.

30991

30992

The *qsub* utility shall set the *Account\_Name* attribute of the batch job to the value of the *account\_string* option-argument.

30993

30994

If the **-A** option is not presented to the *qsub* utility, the utility shall omit the *Account\_Name* attribute from the attributes of the batch job.

30995

**-c interval** Define whether the batch job should be checkpointed, and if so, how often.

30996

30997

The *qsub* utility shall accept a value for the *interval* option-argument that is one of the following:

30998

30999

**n** No checkpointing shall be performed on the batch job (NO\_CHECKPOINT).

31000

31001

**s** Checkpointing shall be performed only when the batch server is shut down (CHECKPOINT\_AT\_SHUTDOWN).

31002

31003

31004

**c** Automatic periodic checkpointing shall be performed at the *Minimum\_Cpu\_Interval* attribute of the batch queue, in units of CPU minutes (CHECKPOINT\_AT\_MIN\_CPU\_INTERVAL).

31005

31006

31007

31008

**c=minutes** Automatic periodic checkpointing shall be performed every *minutes* of CPU time, or every *Minimum\_Cpu\_Interval* minutes, whichever is greater. The *minutes* argument shall conform to the syntax for unsigned integers and shall be greater than zero.

31009

31010

The *qsub* utility shall set the *Checkpoint* attribute of the batch job to the value of the *interval* option-argument.

31011 If the **-c** option is not presented to the *qsub* utility, the utility shall set the  
31012 *Checkpoint* attribute of the batch job to the single character '*u*'  
31013 (CHECKPOINT\_UNSPECIFIED).

31014 **-C directive\_prefix**  
31015 Define the prefix that declares a directive to the *qsub* utility within the script.  
31016 The *directive\_prefix* is not a batch job attribute; it affects the behavior of the *qsub*  
31017 utility.  
31018 If the **-C** option is presented to the *qsub* utility, and the value of the *directive\_prefix*  
31019 option-argument is the null string, the utility shall not scan the script file for  
31020 directives. If the **-C** option is not presented to the *qsub* utility, then the value of the  
31021 *PBS\_DPREFIX* environment variable is used. If the environment variable is not  
31022 defined, then #PBS encoded in the portable character set is the default.

31023 **-e path\_name** Define the path to be used for the standard error stream of the batch job.  
31024 The *qsub* utility shall accept a *path\_name* option-argument which can be preceded  
31025 by a host name element of the form *hostname*:.  
31026 If the *path\_name* option-argument constitutes an absolute pathname, the *qsub*  
31027 utility shall set the *Error\_Path* attribute of the batch job to the value of the  
31028 *path\_name* option-argument.  
31029 If the *path\_name* option-argument constitutes a relative pathname and no host  
31030 name element is specified, the *qsub* utility shall set the *Error\_Path* attribute of the  
31031 batch job to the value of the absolute pathname derived by expanding the  
31032 *path\_name* option-argument relative to the current directory of the process  
31033 executing *qsub*.  
31034 If the *path\_name* option-argument constitutes a relative pathname and a host name  
31035 element is specified, the *qsub* utility shall set the *Error\_Path* attribute of the batch  
31036 job to the value of the *path\_name* option-argument without expansion. The host  
31037 name element shall be included.  
31038 If the *path\_name* option-argument does not include a host name element, the *qsub*  
31039 utility shall prefix the pathname with *hostname*:;, where *hostname* is the name of the  
31040 host upon which the *qsub* utility is being executed.  
31041 If the **-e** option is not presented to the *qsub* utility, the utility shall set the  
31042 *Error\_Path* attribute of the batch job to the host name and path of the current  
31043 directory of the submitting process and the default filename.  
31044 The default filename for standard error has the following format:  
31045 *job\_name.esequence\_number*  
31046 **-h** Specify that a USER hold is applied to the batch job.  
31047 The *qsub* utility shall set the value of the *Hold\_Types* attribute of the batch job to the  
31048 value USER.  
31049 If the **-h** option is not presented to the *qsub* utility, the utility shall set the  
31050 *Hold\_Types* attribute of the batch job to the value NO\_HOLD.  
31051 **-j join\_list** Define which streams of the batch job are to be merged. The *qsub -j* option shall  
31052 accept a value for the *join\_list* option-argument that is a string of alphanumeric  
31053 characters in the portable character set (see the Base Definitions volume of  
31054 IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

31055 The *qsub* utility shall accept a *join\_list* option-argument that consists of one or  
31056 more of the characters 'e' and 'o', or the single character 'n'.

31057 All of the other batch job output streams specified will be merged into the output  
31058 stream represented by the character listed first in the *join\_list* option-argument.

31059 For each unique character in the *join\_list* option-argument, the *qsub* utility shall  
31060 add a value to the *Join\_Path* attribute of the batch job as follows, each representing  
31061 a different batch job stream to join:

- 31062 e The standard error of the batch job (JOIN\_STD\_ERROR).
- 31063 o The standard output of the batch job (JOIN\_STD\_OUTPUT).

31064 An existing *Join\_Path* attribute can be cleared by the following join type:

31065 n NO\_JOIN

31066 If 'n' is specified, then no files are joined. The *qsub* utility shall consider it an error  
31067 if any join type other than 'n' is combined with join type 'n'.

31068 Strictly conforming applications shall not repeat any of the characters 'e', 'o', or  
31069 'n' within the *join\_list* option-argument. The *qsub* utility shall permit the  
31070 repetition of characters, but shall not assign additional meaning to the repeated  
31071 characters.

31072 An implementation may define other join types. The conformance document for an  
31073 implementation shall describe any additional batch job streams, how they are  
31074 specified, their internal behavior, and how they affect the behavior of the utility.

31075 If the **-j** option is not presented to the *qsub* utility, the utility shall set the value of  
31076 the *Join\_Path* attribute of the batch job to NO\_JOIN.

31077 **-k** *keep\_list* Define which output of the batch job to retain on the execution host.

31078 The *qsub -k* option shall accept a value for the *keep\_list* option-argument that is a  
31079 string of alphanumeric characters in the portable character set (see the Base  
31080 Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

31081 The *qsub* utility shall accept a *keep\_list* option-argument that consists of one or  
31082 more of the characters 'e' and 'o', or the single character 'n'.

31083 For each unique character in the *keep\_list* option-argument, the *qsub* utility shall  
31084 add a value to the *Keep\_Files* attribute of the batch job as follows, each representing  
31085 a different batch job stream to keep:

- 31086 e The standard error of the batch job (KEEP\_STD\_ERROR).
- 31087 o The standard output of the batch job (KEEP\_STD\_OUTPUT).

31088 If both 'e' and 'o' are specified, then both files are retained. An existing  
31089 *Keep\_Files* attribute can be cleared by the following keep type:

31090 n NO\_KEEP

31091 If 'n' is specified, then no files are retained. The *qsub* utility shall consider it an  
31092 error if any keep type other than 'n' is combined with keep type 'n'.

31093 Strictly conforming applications shall not repeat any of the characters 'e', 'o', or  
31094 'n' within the *keep\_list* option-argument. The *qsub* utility shall permit the  
31095 repetition of characters, but shall not assign additional meaning to the repeated  
31096 characters.

31097 An implementation may define other keep types. The conformance document for  
31098 an implementation shall describe any additional keep types, how they are  
31099 specified, their internal behavior, and how they affect the behavior of the utility. If  
31100 the **-k** option is not presented to the *qsub* utility, the utility shall set the *Keep\_Files*  
31101 attribute of the batch job to the value NO\_KEEP.

31102 **-m mail\_options**

31103 Define the points in the execution of the batch job at which the batch server that  
31104 manages the batch job shall send mail about a change in the state of the batch job.

31105 The *qsub -m* option shall accept a value for the *mail\_options* option-argument that  
31106 is a string of alphanumeric characters in the portable character set (see the Base  
31107 Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set).

31108 The *qsub* utility shall accept a value for the *mail\_options* option-argument that is a  
31109 string of one or more of the characters 'e', 'b', and 'a', or the single character  
31110 'n'.

31111 For each unique character in the *mail\_options* option-argument, the *qsub* utility shall  
31112 add a value to the *Mail\_Users* attribute of the batch job as follows, each  
31113 representing a different time during the life of a batch job at which to send mail:

- 31114    e    MAIL\_AT\_EXIT  
31115    b    MAIL\_AT\_BEGINNING  
31116    a    MAIL\_AT\_ABORT

31117 If any of these characters are duplicated in the *mail\_options* option-argument, the  
31118 duplicates shall be ignored.

31119 An existing *Mail\_Points* attribute can be cleared by the following mail type:

- 31120    n    NO\_MAIL

31121 If 'n' is specified, then mail is not sent. The *qsub* utility shall consider it an error if  
31122 any mail type other than 'n' is combined with mail type 'n'.

31123 Strictly conforming applications shall not repeat any of the characters 'e', 'b',  
31124 'a', or 'n' within the *mail\_options* option-argument.

31125 The *qsub* utility shall permit the repetition of characters, but shall not assign  
31126 additional meaning to the repeated characters. An implementation may define  
31127 other mail types. The conformance document for an implementation shall describe  
31128 any additional mail types, how they are specified, their internal behavior, and how  
31129 they affect the behavior of the utility.

31130 If the **-m** option is not presented to the *qsub* utility, the utility shall set the  
31131 *Mail\_Points* attribute to the value MAIL\_AT\_ABORT.

31132 **-M mail\_list** Define the list of users to which a batch server that executes the batch job shall  
31133 send mail, if the server sends mail about the batch job.

31134 The syntax of the *mail\_list* option-argument is unspecified.

31135 If the implementation of the *qsub* utility uses a name service to locate users, the  
31136 utility should accept the syntax used by the name service.

31137 If the implementation of the *qsub* utility does not use a name service to locate  
31138 users, the implementation should accept the following syntax for user names:

31139                   `mail_address[ , mail_address , , . . . ]`

31140                   The interpretation of *mail\_address* is implementation-defined.

31141                   The *qsub* utility shall set the *Mail\_Users* attribute of the batch job to the value of the *mail\_list* option-argument.

31143                   If the **-M** option is not presented to the *qsub* utility, the utility shall place only the user name and host name for the current process in the *Mail\_Users* attribute of the batch job.

31146                  **-N name** Define the name of the batch job.

31147                   The *qsub* **-N** option shall accept a value for the *name* option-argument that is a string of up to 15 alphanumeric characters in the portable character set (see the Base Definitions volume of IEEE Std 1003.1-2001, Section 6.1, Portable Character Set) where the first character is alphabetic.

31151                   The *qsub* utility shall set the value of the *Job\_Name* attribute of the batch job to the value of the *name* option-argument.

31153                   If the **-N** option is not presented to the *qsub* utility, the utility shall set the *Job\_Name* attribute of the batch job to the name of the *script* argument from which the directory specification if any, has been removed.

31156                   If the **-N** option is not presented to the *qsub* utility, and the script is read from standard input, the utility shall set the *Job\_Name* attribute of the batch job to the value STDIN.

31159                  **-o path\_name** Define the path for the standard output of the batch job.

31160                   The *qsub* utility shall accept a *path\_name* option-argument that conforms to the syntax of the *path\_name* element defined in the System Interfaces volume of IEEE Std 1003.1-2001, which can be preceded by a host name element of the form *hostname*:

31164                   If the *path\_name* option-argument constitutes an absolute pathname, the *qsub* utility shall set the *Output\_Path* attribute of the batch job to the value of the *path\_name* option-argument without expansion.

31167                   If the *path\_name* option-argument constitutes a relative pathname and no host name element is specified, the *qsub* utility shall set the *Output\_Path* attribute of the batch job to the pathname derived by expanding the value of the *path\_name* option-argument relative to the current directory of the process executing the *qsub*.

31171                   If the *path\_name* option-argument constitutes a relative pathname and a host name element is specified, the *qsub* utility shall set the *Output\_Path* attribute of the batch job to the value of the *path\_name* option-argument without expansion.

31174                   If the *path\_name* option-argument does not specify a host name element, the *qsub* utility shall prefix the pathname with *hostname*:; where *hostname* is the name of the host upon which the *qsub* utility is executing.

31177                   If the **-o** option is not presented to the *qsub* utility, the utility shall set the *Output\_Path* attribute of the batch job to the host name and path of the current directory of the submitting process and the default filename.

31180                   The default filename for standard output has the following format:

31181                  `job_name.sequence_number`

|       |                          |                                                                                                                                                                                                                                                                                                                                   |
|-------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31182 | <b>-p priority</b>       | Define the priority the batch job should have relative to other batch jobs owned by the batch server.                                                                                                                                                                                                                             |
| 31184 |                          | The <i>qsub</i> utility shall set the <i>Priority</i> attribute of the batch job to the value of the <i>priority</i> option-argument.                                                                                                                                                                                             |
| 31186 |                          | If the <b>-p</b> option is not presented to the <i>qsub</i> utility, the value of the <i>Priority</i> attribute is implementation-defined.                                                                                                                                                                                        |
| 31188 |                          | The <i>qsub</i> utility shall accept a value for the <i>priority</i> option-argument that conforms to the syntax for signed decimal integers, and which is not less than -1 024 and not greater than 1 023.                                                                                                                       |
| 31191 | <b>-q destination</b>    | Define the destination of the batch job.                                                                                                                                                                                                                                                                                          |
| 31193 |                          | The destination is not a batch job attribute; it determines the batch server, and possibly the batch queue, to which the <i>qsub</i> utility batch queues the batch job.                                                                                                                                                          |
| 31195 |                          | The <i>qsub</i> utility shall submit the script to the batch server named by the <i>destination</i> option-argument or the server that owns the batch queue named in the <i>destination</i> option-argument.                                                                                                                      |
| 31198 |                          | The <i>qsub</i> utility shall accept an option-argument for the <b>-q</b> option that conforms to the syntax for a destination (see Section 3.3.2 (on page 123)).                                                                                                                                                                 |
| 31200 |                          | If the <b>-q</b> option is not presented to the <i>qsub</i> utility, the <i>qsub</i> utility shall submit the batch job to the default destination. The mechanism for determining the default destination is implementation-defined.                                                                                              |
| 31203 | <b>-r y n</b>            | Define whether the batch job is rerunnable.                                                                                                                                                                                                                                                                                       |
| 31204 |                          | If the value of the option-argument is <i>y</i> , the <i>qsub</i> utility shall set the <i>Rerunable</i> attribute of the batch job to TRUE.                                                                                                                                                                                      |
| 31206 |                          | If the value of the option-argument is <i>n</i> , the <i>qsub</i> utility shall set the <i>Rerunable</i> attribute of the batch job to FALSE.                                                                                                                                                                                     |
| 31208 |                          | If the <b>-r</b> option is not presented to the <i>qsub</i> utility, the utility shall set the <i>Rerunable</i> attribute of the batch job to TRUE.                                                                                                                                                                               |
| 31210 | <b>-S path_name_list</b> | Define the pathname to the shell under which the batch job is to execute.                                                                                                                                                                                                                                                         |
| 31212 |                          | The <i>qsub</i> utility shall accept a <i>path_name_list</i> option-argument that conforms to the following syntax:                                                                                                                                                                                                               |
| 31214 |                          | <i>pathname[@host][,, pathname[@host], , . . .]</i>                                                                                                                                                                                                                                                                               |
| 31215 |                          | The <i>qsub</i> utility shall allow only one pathname for a given host name. The <i>qsub</i> utility shall allow only one pathname that is missing a corresponding host name.                                                                                                                                                     |
| 31217 |                          | The <i>qsub</i> utility shall add a value to the <i>Shell_Path_List</i> attribute of the batch job for each entry in the <i>path_name_list</i> option-argument.                                                                                                                                                                   |
| 31219 |                          | If the <b>-S</b> option is not presented to the <i>qsub</i> utility, the utility shall set the <i>Shell_Path_List</i> attribute of the batch job to the null string.                                                                                                                                                              |
| 31221 |                          | The conformance document for an implementation shall describe the mechanism used to set the default shell and determine the current value of the default shell. An implementation shall provide a means for the installation to set the default shell to the login shell of the user under which the batch job is to execute. See |

31225                   Section 3.3.3 (on page 123) for a means of removing *keyword=value* (and  
31226                   *value@keyword*) pairs and other general rules for list-oriented batch job attributes.

**-u user\_list** Define the user name under which the batch job is to execute.

The *qsub* utility shall accept a *user\_list* option-argument that conforms to the following syntax:

**31230**                    *username[@host][,,username[@host],,, ...]*

31231 The *qsub* utility shall accept only one user name that is missing a corresponding  
31232 host name. The *qsub* utility shall accept only one user name per named host.

The *qsub* utility shall add a value to the *User\_List* attribute of the batch job for each entry in the *user\_list* option-argument.

If the **-u** option is not presented to the *qsub* utility, the utility shall set the *User\_List* attribute of the batch job to the user name from which the utility is executing. See Section 3.3.3 (on page 123) for a means of removing *keyword=value* (and *value@keyword*) pairs and other general rules for list-oriented batch job attributes.

**31239**                   **-v variable\_list**

**31240** Add to the list of variables that are exported to the session leader of the batch job.

A *variable\_list* is a set of strings of either the form <variable> or <variable=value>, delimited by commas.

31243 If the **-v** option is presented to the *qsub* utility, the utility shall also add, to the  
31244 environment *Variable\_List* attribute of the batch job, every variable named in the  
31245 environment *variable\_list* option-argument and, optionally, values of specified  
31246 variables.

31247 If a value is not provided on the command line, the *qsub* utility shall set the value  
31248 of each variable in the environment *Variable\_List* attribute of the batch job to the  
31249 value of the corresponding environment variable for the process in which the  
31250 utility is executing; see Table 4-18 (on page 801).

31251 A conforming application shall not repeat a variable in the environment  
31252 *variable\_list* option-argument.

31253 The *qsub* utility shall not repeat a variable in the environment *Variable\_List*  
31254 attribute of the batch job. See Section 3.3.3 (on page 123) for a means of removing  
31255 *keyword=value* (and *value@keyword*) pairs and other general rules for list-oriented  
31256 batch job attributes.

31257           **-V**           Specify that all of the environment variables of the process are exported to the context of the batch job.  
31258

31259 The *qsub* utility shall place every environment variable in the process in which the  
31260 utility is executing in the list and shall set the value of each variable in the attribute  
31261 to the value of that variable in the process.

31262           **-z**           Specify that the utility does not write the batch *job\_identifier* of the created batch job to standard output.  
31263

31264 If the **-z** option is presented to the *qsub* utility, the utility shall not write the batch  
31265 *job\_identifier* of the created batch job to standard output.

If the **-z** option is not presented to the *qsub* utility, the utility shall write the identifier of the created batch job to standard output.

**31268 OPERANDS**

31269 The *qsub* utility shall accept a *script* operand that indicates the path to the script of the batch job.  
31270 If the *script* operand is not presented to the *qsub* utility, or if the operand is the single-character  
31271 string ' - ', the utility shall read the script from standard input.  
31272 If the script represents a partial path, the *qsub* utility shall expand the path relative to the current  
31273 directory of the process executing the utility.

**31274 STDIN**

31275 The *qsub* utility reads the script of the batch job from standard input if the script operand is  
31276 omitted or is the single character ' - '.

**31277 INPUT FILES**

31278 In addition to binding the file indicated by the *script* operand to the batch job, the *qsub* utility  
31279 reads the script file and acts on directives in the script.

**31280 ENVIRONMENT VARIABLES**

31281 The following environment variables shall affect the execution of *qsub*:

31282 *LANG* Provide a default value for the internationalization variables that are unset or null.  
31283 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31284 Internationalization Variables for the precedence of internationalization variables  
31285 used to determine the values of locale categories.)

31286 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
31287 internationalization variables.

31288 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
31289 characters (for example, single-byte as opposed to multi-byte characters in  
31290 arguments).

**31291 *LC\_MESSAGES***

31292 Determine the locale that should be used to affect the format and contents of  
31293 diagnostic messages written to standard error.

31294 *LOGNAME* Determine the login name of the user.

31295 *PBS\_DPREFIX*  
31296 Determine the default prefix for directives within the script.

31297 *SHELL* Determine the pathname of the preferred command language interpreter of the  
31298 user.

31299 *TZ* Determine the timezone used to interpret the *date-time* option-argument. If *TZ* is  
31300 unset or null, an unspecified default timezone shall be used.

**31301 ASYNCHRONOUS EVENTS**

31302 Once created, a batch job exists until it exits, aborts, or is deleted.

31303 After a batch job is created by the *qsub* utility, batch servers might route, execute, modify, or  
31304 delete the batch job.

**31305 STDOUT**

31306 The *qsub* utility writes the batch *job\_identifier* assigned to the batch job to standard output, unless  
31307 the *-z* option is specified.

**31308 STDERR**

31309 The standard error shall be used only for diagnostic messages.

**31310 OUTPUT FILES**

31311 None.

**31312 EXTENDED DESCRIPTION****31313 Script Preservation**

31314 The *qsub* utility shall make the script available to the server executing the batch job in such a way  
31315 that the server executes the script as it exists at the time of submission.

31316 The *qsub* utility can send a copy of the script to the server with the *Queue Job Request* or store a  
31317 temporary copy of the script in a location specified to the server.

**31318 Option Specification**

31319 A script can contain directives to the *qsub* utility.

31320 The *qsub* utility shall scan the lines of the script for directives, skipping blank lines, until the first  
31321 line that begins with a string other than the directive string; if directives occur on subsequent  
31322 lines, the utility shall ignore those directives.

31323 Lines are separated by a <newline>. If the first line of the script begins with "#!" or a colon  
31324 (' : '), then it is skipped. The *qsub* utility shall process a line in the script as a directive if and  
31325 only if the string of characters from the first non-white-space character on the line until the first  
31326 <space> or <tab> on the line match the directive prefix. If a line in the script contains a directive  
31327 and the final characters of the line are backslash ('\') and <newline>, then the next line shall be  
31328 interpreted as a continuation of that directive.

31329 The *qsub* utility shall process the options and option-arguments contained on the directive prefix  
31330 line using the same syntax as if the options were input on the *qsub* utility.

31331 The *qsub* utility shall continue to process a directive prefix line until after a <newline> is  
31332 encountered. An implementation may ignore lines which, according to the syntax of the shell  
31333 that will interpret the script, are comments. An implementation shall describe in the  
31334 conformance document the format of any shell comments that it will recognize.

31335 If an option is present in both a directive and the arguments to the *qsub* utility, the utility shall  
31336 ignore the option and the corresponding option-argument, if any, in the directive.

31337 If an option that is present in the directive is not present in the arguments to the *qsub* utility, the  
31338 utility shall process the option and the option-argument, if any.

31339 In order of preference, the *qsub* utility shall select the directive prefix from one of the following  
31340 sources:

- If the **-C** option is presented to the utility, the value of the *directive\_prefix* option-argument
- If the environment variable *PBS\_DPREFIX* is defined, the value of that variable
- The four-character string "#PBS" encoded in the portable character set

31344 If the **-C** option is present in the script file it shall be ignored.

**31345 EXIT STATUS**

31346 The following exit values shall be returned:

31347 0 Successful completion.

31348 >0 An error occurred.

### 31349 CONSEQUENCES OF ERRORS

31350 Default.

### 31351 APPLICATION USAGE

31352 None.

### 31353 EXAMPLES

31354 None.

### 31355 RATIONALE

31356 The *qsub* utility allows users to create a batch job that will process the script specified as the  
31357 operand of the utility.

31358 The options of the *qsub* utility allow users to control many aspects of the queuing and execution  
31359 of a batch job.

31360 The **-a** option allows users to designate the time after which the batch job will become eligible to  
31361 run. By specifying an execution time, users can take advantage of resources at off-peak hours,  
31362 synchronize jobs with chronologically predictable events, and perhaps take advantage of off-  
31363 peak pricing of computing time. For these reasons and others, a timing option is existing practice  
31364 on the part of almost every batch system, including NQS.

31365 The **-A** option allows users to specify the account that will be charged for the batch job. Support  
31366 for account is not mandatory for conforming batch servers.

31367 The **-C** option allows users to prescribe the prefix for directives within the script file. The default  
31368 prefix "#PBS" may be inappropriate if the script will be interpreted with an alternate shell, as  
31369 specified by the **-S** option.

31370 The **-c** option allows users to establish the checkpointing interval for their jobs. A checkpointing  
31371 system, which is not defined by this volume of IEEE Std 1003.1-2001, allows recovery of a batch  
31372 job at the most recent checkpoint in the event of a crash. Checkpointing is typically used for jobs  
31373 that consume expensive computing time or must meet a critical schedule. Users should be  
31374 allowed to make the tradeoff between the overhead of checkpointing and the risk to the timely  
31375 completion of the batch job; therefore, this volume of IEEE Std 1003.1-2001 provides the  
31376 checkpointing interval option. Support for checkpointing is optional for batch servers.

31377 The **-e** option allows users to redirect the standard error streams of their jobs to a non-default  
31378 path. For example, if the submitted script generally produces a great deal of useless error output,  
31379 a user might redirect the standard error output to the null device. Or, if the file system holding  
31380 the default location (the home directory of the user) has too little free space, the user might  
31381 redirect the standard error stream to a file in another file system.

31382 The **-h** option allows users to create a batch job that is held until explicitly released. The ability  
31383 to create a held job is useful when some external event must complete before the batch job can  
31384 execute. For example, the user might submit a held job and release it when the system load has  
31385 dropped.

31386 The **-j** option allows users to merge the standard error of a batch job into its standard output  
31387 stream, which has the advantage of showing the sequential relationship between output and  
31388 error messages.

31389 The **-m** option allows users to designate those points in the execution of a batch job at which  
31390 mail will be sent to the submitting user, or to the account(s) indicated by the **-M** option. By  
31391 requesting mail notification at points of interest in the life of a job, the submitting user, or other  
31392 designated users, can track the progress of a batch job.

31393        The **-N** option allows users to associate a name with the batch job. The job name in no way  
31394        affects the processing of the batch job, but rather serves as a mnemonic handle for users. For  
31395        example, the batch job name can help the user distinguish between multiple jobs listed by the  
31396        *qstat* utility.

31397        The **-o** option allows users to redirect the standard output stream. A user might, for example,  
31398        wish to redirect to the null device the standard output stream of a job that produces copious yet  
31399        superfluous output.

31400        The **-P** option allows users to designate the relative priority of a batch job for selection from a  
31401        queue.

31402        The **-q** option allows users to specify an initial queue for the batch job. If the user specifies a  
31403        routing queue, the batch server routes the batch job to another queue for execution or further  
31404        routing. If the user specifies a non-routing queue, the batch server of the queue eventually  
31405        executes the batch job.

31406        The **-r** option allows users to control whether the submitted job will be rerun if the controlling  
31407        batch node fails during execution of the batch job. The **-r** option likewise allows users to  
31408        indicate whether or not the batch job is eligible to be rerun by the *qrerun* utility. Some jobs cannot  
31409        be correctly rerun because of changes they make in the state of databases or other aspects of  
31410        their environment. This volume of IEEE Std 1003.1-2001 specifies that the default, if the **-r**  
31411        option is not presented to the utility, will be that the batch job cannot be rerun, since the result of  
31412        rerunning a non-runnable job might be catastrophic.

31413        The **-S** option allows users to specify the program (usually a shell) that will be invoked to  
31414        process the script of the batch job. This option has been modified to allow a list of shell names  
31415        and locations associated with different hosts.

31416        The **-u** option is useful when the submitting user is authorized to use more than one account on  
31417        a given host, in which case the **-u** option allows the user to select from among those accounts.  
31418        The option-argument is a list of user-host pairs, so that the submitting user can provide different  
31419        user identifiers for different nodes in the event the batch job is routed. The **-u** option provides a  
31420        lot of flexibility to accommodate sites with complex account structures. Users that have the  
31421        same user identifier on all the hosts they are authorized to use will not need to use the **-u** option.

31422        The **-V** option allows users to export all their current environment variables, as of the time the  
31423        batch job is submitted, to the context of the processes of the batch job.

31424        The **-v** option allows users to export specific environment variables from their current process  
31425        to the processes of the batch job.

31426        The **-z** option allows users to suppress the writing of the batch job identifier to standard output.  
31427        The **-z** option is an existing NQS practice that has been standardized.

31428        Historically, the *qsub* utility has served the batch job-submission function in the NQS system, the  
31429        existing practice on which it is based. Some changes and additions have been made to the *qsub*  
31430        utility in this volume of IEEE Std 1003.1-2001, *vis-a-vis* NQS, as a result of the growing pool of  
31431        experience with distributed batch systems.

31432        The set of features of the *qsub* utility as defined in this volume of IEEE Std 1003.1-2001 appears to  
31433        incorporate all the common existing practice on potentially conforming platforms.

31434 **FUTURE DIRECTIONS**

31435        None.

31436 **SEE ALSO**31437        Chapter 3 (on page 101), *qrerun*, *qstat*, *touch*31438 **CHANGE HISTORY**

31439        Derived from IEEE Std 1003.2d-1994.

31440 **Issue 6**31441        The **-l** option has been removed as there is no portable description of the resources that are  
31442        allowed or required by the batch job.

**31443 NAME**

31444        *read* — read a line from standard input

**31445 SYNOPSIS**

31446        *read [-r] var...*

**31447 DESCRIPTION**

31448        The *read* utility shall read a single line from standard input.

31449        By default, unless the *-r* option is specified, backslash ('\\') shall act as an escape character, as  
31450        described in Section 2.2.1 (on page 30). If standard input is a terminal device and the invoking  
31451        shell is interactive, *read* shall prompt for a continuation line when:

- 31452        • The shell reads an input line ending with a backslash, unless the *-r* option is specified.
- 31453        • A here-document is not terminated after a <newline> is entered.

31454        The line shall be split into fields as in the shell (see Section 2.6.5 (on page 42)); the first field shall  
31455        be assigned to the first variable *var*, the second field to the second variable *var*, and so on. If  
31456        there are fewer *var* operands specified than there are fields, the leftover fields and their  
31457        intervening separators shall be assigned to the last *var*. If there are fewer fields than *vars*, the  
31458        remaining *vars* shall be set to empty strings.

31459        The setting of variables specified by the *var* operands shall affect the current shell execution  
31460        environment; see Section 2.12 (on page 61). If it is called in a subshell or separate utility  
31461        execution environment, such as one of the following:

31462        (read foo)  
31463        nohup read ...  
31464        find . -exec read ... \\;

31465        it shall not affect the shell variables in the caller's environment.

**31466 OPTIONS**

31467        The *read* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
31468        12.2, Utility Syntax Guidelines.

31469        The following option is supported:

31470        **-r**        Do not treat a backslash character in any special way. Consider each backslash to  
31471        be part of the input line.

**31472 OPERANDS**

31473        The following operand shall be supported:

31474        *var*        The name of an existing or nonexisting shell variable.

**31475 STDIN**

31476        The standard input shall be a text file.

**31477 INPUT FILES**

31478        None.

**31479 ENVIRONMENT VARIABLES**

31480        The following environment variables shall affect the execution of *read*:

31481        *IFS*        Determine the internal field separators used to delimit fields; see Section 2.5.3 (on  
31482        page 34).

31483        *LANG*      Provide a default value for the internationalization variables that are unset or null.  
31484        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31485        Internationalization Variables for the precedence of internationalization variables

|           |                                                     |                                                                                                                                                                                                                                                 |
|-----------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31486     | used to determine the values of locale categories.) |                                                                                                                                                                                                                                                 |
| 31487     | <i>LC_ALL</i>                                       | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                        |
| 31488     |                                                     |                                                                                                                                                                                                                                                 |
| 31489     | <i>LC_CTYPE</i>                                     | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                       |
| 31490     |                                                     |                                                                                                                                                                                                                                                 |
| 31491     |                                                     |                                                                                                                                                                                                                                                 |
| 31492     | <i>LC_MESSAGES</i>                                  | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                    |
| 31493     |                                                     |                                                                                                                                                                                                                                                 |
| 31494     |                                                     |                                                                                                                                                                                                                                                 |
| 31495 XSI | <i>NLSPATH</i>                                      | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                           |
| 31496     | <i>PS2</i>                                          | Provide the prompt string that an interactive shell shall write to standard error when a line ending with a backslash is read and the <i>-r</i> option was not specified, or if a here-document is not terminated after a <newline> is entered. |
| 31497     |                                                     |                                                                                                                                                                                                                                                 |
| 31498     |                                                     |                                                                                                                                                                                                                                                 |
| 31499     | <b>ASYNCHRONOUS EVENTS</b>                          |                                                                                                                                                                                                                                                 |
| 31500     |                                                     | Default.                                                                                                                                                                                                                                        |
| 31501     | <b>STDOUT</b>                                       |                                                                                                                                                                                                                                                 |
| 31502     |                                                     | Not used.                                                                                                                                                                                                                                       |
| 31503     | <b>STDERR</b>                                       |                                                                                                                                                                                                                                                 |
| 31504     |                                                     | The standard error shall be used for diagnostic messages and prompts for continued input.                                                                                                                                                       |
| 31505     | <b>OUTPUT FILES</b>                                 |                                                                                                                                                                                                                                                 |
| 31506     |                                                     | None.                                                                                                                                                                                                                                           |
| 31507     | <b>EXTENDED DESCRIPTION</b>                         |                                                                                                                                                                                                                                                 |
| 31508     |                                                     | None.                                                                                                                                                                                                                                           |
| 31509     | <b>EXIT STATUS</b>                                  |                                                                                                                                                                                                                                                 |
| 31510     |                                                     | The following exit values shall be returned:                                                                                                                                                                                                    |
| 31511     |                                                     | 0 Successful completion.                                                                                                                                                                                                                        |
| 31512     |                                                     | >0 End-of-file was detected or an error occurred.                                                                                                                                                                                               |
| 31513     | <b>CONSEQUENCES OF ERRORS</b>                       |                                                                                                                                                                                                                                                 |
| 31514     |                                                     | Default.                                                                                                                                                                                                                                        |
| 31515     | <b>APPLICATION USAGE</b>                            |                                                                                                                                                                                                                                                 |
| 31516     |                                                     | The <i>-r</i> option is included to enable <i>read</i> to subsume the purpose of the <i>line</i> utility, which is not included in IEEE Std 1003.1-2001.                                                                                        |
| 31517     |                                                     |                                                                                                                                                                                                                                                 |
| 31518     |                                                     | The results are undefined if an end-of-file is detected following a backslash at the end of a line when <i>-r</i> is not specified.                                                                                                             |
| 31519     |                                                     |                                                                                                                                                                                                                                                 |
| 31520     | <b>EXAMPLES</b>                                     |                                                                                                                                                                                                                                                 |
| 31521     |                                                     | The following command:                                                                                                                                                                                                                          |
| 31522     |                                                     | while read -r xx yy                                                                                                                                                                                                                             |
| 31523     |                                                     | do                                                                                                                                                                                                                                              |
| 31524     |                                                     | printf "%s %s\n" "\$yy" "\$xx"                                                                                                                                                                                                                  |
| 31525     |                                                     | done < <i>input_file</i>                                                                                                                                                                                                                        |
| 31526     |                                                     | prints a file with the first field of each line moved to the end of the line.                                                                                                                                                                   |

**31527 RATIONALE**

31528       The *read* utility historically has been a shell built-in. It was separated off into its own utility to  
31529       take advantage of the richer description of functionality introduced by this volume of  
31530       IEEE Std 1003.1-2001.

31531       Since *read* affects the current shell execution environment, it is generally provided as a shell  
31532       regular built-in. If it is called in a subshell or separate utility execution environment, such as one  
31533       of the following:

31534           (read foo)  
31535           nohup read ...  
31536           find . -exec read ... \\;

31537       it does not affect the shell variables in the environment of the caller.

**31538 FUTURE DIRECTIONS**

31539       None.

**31540 SEE ALSO**

31541       Chapter 2 (on page 29)

**31542 CHANGE HISTORY**

31543       First released in Issue 2.

## 31544 NAME

31545        renice — set nice values of running processes

## 31546 SYNOPSIS

31547 UP      `renice -n increment [-g | -p | -u] ID ...`

31548

## 31549 DESCRIPTION

31550        The *renice* utility shall request that the nice values (see the Base Definitions volume of  
31551        IEEE Std 1003.1-2001, Section 3.239, Nice Value) of one or more running processes be changed.  
31552        By default, the applicable processes are specified by their process IDs. When a process group is  
31553        specified (see **-g**), the request shall apply to all processes in the process group.

31554        The nice value shall be bounded in an implementation-defined manner. If the requested  
31555        *increment* would raise or lower the nice value of the executed utility beyond implementation-  
31556        defined limits, then the limit whose value was exceeded shall be used.

31557        When a user is *reniced*, the request applies to all processes whose saved set-user-ID matches the  
31558        user ID corresponding to the user.

31559        Regardless of which options are supplied or any other factor, *renice* shall not alter the nice values  
31560        of any process unless the user requesting such a change has appropriate privileges to do so for  
31561        the specified process. If the user lacks appropriate privileges to perform the requested action, the  
31562        utility shall return an error status.

31563        The saved set-user-ID of the user's process shall be checked instead of its effective user ID when  
31564        *renice* attempts to determine the user ID of the process in order to determine whether the user  
31565        has appropriate privileges.

## 31566 OPTIONS

31567        The *renice* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
31568        12.2, Utility Syntax Guidelines.

31569        The following options shall be supported:

31570        **-g**        Interpret all operands as unsigned decimal integer process group IDs.

31571        **-n increment**        Specify how the nice value of the specified process or processes is to be adjusted.  
31572        The *increment* option-argument is a positive or negative decimal integer that shall  
31573        be used to modify the nice value of the specified process or processes.

31574        Positive *increment* values shall cause a lower nice value. Negative *increment* values  
31575        may require appropriate privileges and shall cause a higher nice value.

31576        **-p**        Interpret all operands as unsigned decimal integer process IDs. The **-p** option is  
31577        the default if no options are specified.

31578        **-u**        Interpret all operands as users. If a user exists with a user name equal to the  
31579        operand, then the user ID of that user is used in further processing. Otherwise, if  
31580        the operand represents an unsigned decimal integer, it shall be used as the numeric  
31581        user ID of the user.

## 31582 OPERANDS

31583        The following operands shall be supported:

31584        **ID**        A process ID, process group ID, or user name/user ID, depending on the option  
31585        selected.

**31586 STDIN**

31587 Not used.

**31588 INPUT FILES**

31589 None.

**31590 ENVIRONMENT VARIABLES**

31591 The following environment variables shall affect the execution of *renice*:

31592 **LANG** Provide a default value for the internationalization variables that are unset or null.  
31593 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31594 Internationalization Variables for the precedence of internationalization variables  
31595 used to determine the values of locale categories.)

31596 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
31597 internationalization variables.

31598 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
31599 characters (for example, single-byte as opposed to multi-byte characters in  
31600 arguments).

**31601 LC\_MESSAGES**

31602 Determine the locale that should be used to affect the format and contents of  
31603 diagnostic messages written to standard error.

31604 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**31605 ASYNCHRONOUS EVENTS**

31606 Default.

**31607 STDOUT**

31608 Not used.

**31609 STDERR**

31610 The standard error shall be used only for diagnostic messages.

**31611 OUTPUT FILES**

31612 None.

**31613 EXTENDED DESCRIPTION**

31614 None.

**31615 EXIT STATUS**

31616 The following exit values shall be returned:

31617 0 Successful completion.

31618 >0 An error occurred.

**31619 CONSEQUENCES OF ERRORS**

31620 Default.

## 31621 APPLICATION USAGE

31622 None.

## 31623 EXAMPLES

- 31624 1. Adjust the nice value so that process IDs 987 and 32 would have a lower nice value:

31625 `renice -n 5 -p 987 32`

- 31626 2. Adjust the nice value so that group IDs 324 and 76 would have a higher nice value, if the
- 
- 31627 user has the appropriate privileges to do so:

31628 `renice -n -4 -g 324 76`

- 31629 3. Adjust the nice value so that numeric user ID 8 and user
- sas**
- would have a lower nice
- 
- 31630 value:

31631 `renice -n 4 -u 8 sas`

31632 Useful nice value increments on historical systems include 19 or 20 (the affected processes run  
31633 only when nothing else in the system attempts to run) and any negative number (to make  
31634 processes run faster).

## 31635 RATIONALE

31636 The *gid*, *pid*, and *user* specifications do not fit either the definition of operand or option-  
31637 argument. However, for clarity, they have been included in the OPTIONS section, rather than  
31638 the OPERANDS section.

31639 The definition of nice value is not intended to suggest that all processes in a system have  
31640 priorities that are comparable. Scheduling policy extensions such as the realtime priorities in the  
31641 System Interfaces volume of IEEE Std 1003.1-2001 make the notion of a single underlying  
31642 priority for all scheduling policies problematic. Some implementations may implement the *nice*-  
31643 related features to affect all processes on the system, others to affect just the general time-  
31644 sharing activities implied by this volume of IEEE Std 1003.1-2001, and others may have no effect  
31645 at all. Because of the use of “implementation-defined” in *nice* and *renice*, a wide range of  
31646 implementation strategies are possible.

31647 Originally, this utility was written in the historical manner, using the term “nice value”. This  
31648 was always a point of concern with users because it was never intuitively obvious what this  
31649 meant. With a newer version of *renice*, which used the term “system scheduling priority”, it was  
31650 hoped that novice users could better understand what this utility was meant to do. Also, it  
31651 would be easier to document what the utility was meant to do. Unfortunately, the addition of  
31652 the POSIX realtime scheduling capabilities introduced the concepts of process and thread  
31653 scheduling priorities that were totally unaffected by the *nice*/*renice* utilities or the  
31654 *nice()*/*setpriority()* functions. Continuing to use the term “system scheduling priority” would  
31655 have incorrectly suggested that these utilities and functions were indeed affecting these realtime  
31656 priorities. It was decided to revert to the historical term “nice value” to reference this unrelated  
31657 process attribute.

31658 Although this utility has use by system administrators (and in fact appears in the system  
31659 administration portion of the BSD documentation), the standard developers considered that it  
31660 was very useful for individual end users to control their own processes.

## 31661 FUTURE DIRECTIONS

31662 None.

31663 **SEE ALSO**

31664       *nice*

31665 **CHANGE HISTORY**

31666       First released in Issue 4.

31667 **Issue 5**

31668       In the SYNOPSIS, an ellipsis is added to the **-u** option in all three obsolescent forms.

31669 **Issue 6**

31670       This utility is marked as part of the User Portability Utilities option.

31671       The APPLICATION USAGE section is added.

31672       The obsolescent forms of the SYNOPSIS are removed.

31673       Text previously conditional on **POSIX\_SAVED\_IDS** is mandatory in this issue. This is a FIPS requirement.  
31674

## 31675 NAME

31676        rm — remove directory entries

## 31677 SYNOPSIS

31678        `rm [-f|Rr] file...`

## 31679 DESCRIPTION

31680        The *rm* utility shall remove the directory entry specified by each *file* argument.31681        If either of the files dot or dot-dot are specified as the basename portion of an operand (that is, 31682        the final pathname component), *rm* shall write a diagnostic message to standard error and do 31683        nothing more with such operands.31684        For each *file* the following steps shall be taken:31685        1. If the *file* does not exist:

- 31686            a. If the
- f**
- option is not specified,
- rm*
- shall write a diagnostic message to standard error.
- 
- 31687            b. Go on to any remaining
- files*
- .

31688        2. If *file* is of type directory, the following steps shall be taken:

- 31689            a. If neither the
- R**
- option nor the
- r**
- option is specified,
- rm*
- shall write a diagnostic
- 
- 31690            message to standard error, do nothing more with
- file*
- , and go on to any remaining
- 
- 31691            files.

- 31692            b. If the
- f**
- option is not specified, and either the permissions of
- file*
- do not permit
- 
- 31693            writing and the standard input is a terminal or the
- i**
- option is specified,
- rm*
- shall
- 
- 31694            write a prompt to standard error and read a line from the standard input. If the
- 
- 31695            response is not affirmative,
- rm*
- shall do nothing more with the current file and go on
- 
- 31696            to any remaining files.

- 31697            c. For each entry contained in
- file*
- , other than dot or dot-dot, the four steps listed here (1
- 
- 31698            to 4) shall be taken with the entry as if it were a
- file*
- operand. The
- rm*
- utility shall not
- 
- 31699            traverse directories by following symbolic links into other parts of the hierarchy, but
- 
- 31700            shall remove the links themselves.

- 31701            d. If the
- i**
- option is specified,
- rm*
- shall write a prompt to standard error and read a line
- 
- 31702            from the standard input. If the response is not affirmative,
- rm*
- shall do nothing more
- 
- 31703            with the current file, and go on to any remaining files.

- 31704        3. If
- file*
- is not of type directory, the
- f**
- option is not specified, and either the permissions of
- 
- 31705
- file*
- do not permit writing and the standard input is a terminal or the
- i**
- option is specified,
- 
- 31706
- rm*
- shall write a prompt to the standard error and read a line from the standard input. If the
- 
- 31707            response is not affirmative,
- rm*
- shall do nothing more with the current file and go on to any
- 
- 31708            remaining files.

- 31709        4. If the current file is a directory,
- rm*
- shall perform actions equivalent to the
- rmdir()*
- function
- 
- 31710            defined in the System Interfaces volume of IEEE Std 1003.1-2001 called with a pathname of
- 
- 31711            the current file used as the
- path*
- argument. If the current file is not a directory,
- rm*
- shall
- 
- 31712            perform actions equivalent to the
- unlink()*
- function defined in the System Interfaces
- 
- 31713            volume of IEEE Std 1003.1-2001 called with a pathname of the current file used as the
- path*
- 
- 31714            argument.

31715        If this fails for any reason, *rm* shall write a diagnostic message to standard error, do  
31716        nothing more with the current file, and go on to any remaining files.31717        The *rm* utility shall be able to descend to arbitrary depths in a file hierarchy, and shall not fail  
31718        due to path length limitations (unless an operand specified by the user exceeds system

31719 limitations).

## 31720 OPTIONS

31721 The **rm** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
31722 Utility Syntax Guidelines.

31723 The following options shall be supported:

- 31724   **-f**       Do not prompt for confirmation. Do not write diagnostic messages or modify the  
31725       exit status in the case of nonexistent operands. Any previous occurrences of the **-i**  
31726       option shall be ignored.
- 31727   **-i**       Prompt for confirmation as described previously. Any previous occurrences of the  
31728       **-f** option shall be ignored.
- 31729   **-R**       Remove file hierarchies. See the DESCRIPTION.
- 31730   **-r**       Equivalent to **-R**.

## 31731 OPERANDS

31732 The following operand shall be supported:

- 31733   **file**      A pathname of a directory entry to be removed.

## 31734 STDIN

31735 The standard input shall be used to read an input line in response to each prompt specified in  
31736 the STDOUT section. Otherwise, the standard input shall not be used.

## 31737 INPUT FILES

31738 None.

## 31739 ENVIRONMENT VARIABLES

31740 The following environment variables shall affect the execution of **rm**:

- 31741   **LANG**      Provide a default value for the internationalization variables that are unset or null.  
31742       (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31743       Internationalization Variables for the precedence of internationalization variables  
31744       used to determine the values of locale categories.)
- 31745   **LC\_ALL**     If set to a non-empty string value, override the values of all the other  
31746       internationalization variables.
- 31747   **LC\_COLLATE**   Determine the locale for the behavior of ranges, equivalence classes, and multi-  
31748       character collating elements used in the extended regular expression defined for  
31750       the **yesexpr** locale keyword in the **LC\_MESSAGES** category.
- 31751   **LC\_CTYPE**    Determine the locale for the interpretation of sequences of bytes of text data as  
31752       characters (for example, single-byte as opposed to multi-byte characters in  
31753       arguments) and the behavior of character classes within regular expressions used  
31754       in the extended regular expression defined for the **yesexpr** locale keyword in the  
31755       **LC\_MESSAGES** category.
- 31756   **LC\_MESSAGES**   Determine the locale for the processing of affirmative responses that should be  
31757       used to affect the format and contents of diagnostic messages written to standard  
31759       error.
- 31760   **XSI\_NLSPATH**   Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

**31761 ASYNCHRONOUS EVENTS**

31762 Default.

**31763 STDOUT**

31764 Not used.

**31765 STDERR**

31766 Prompts shall be written to standard error under the conditions specified in the DESCRIPTION and OPTIONS sections. The prompts shall contain the *file* pathname, but their format is otherwise unspecified. The standard error also shall be used for diagnostic messages.

**31769 OUTPUT FILES**

31770 None.

**31771 EXTENDED DESCRIPTION**

31772 None.

**31773 EXIT STATUS**

31774 The following exit values shall be returned:

31775 0 All of the named directory entries for which *rm* performed actions equivalent to the *rmdir()* or *unlink()* functions were removed.

31777 >0 An error occurred.

**31778 CONSEQUENCES OF ERRORS**

31779 Default.

**31780 APPLICATION USAGE**

31781 The *rm* utility is forbidden to remove the names dot and dot-dot in order to avoid the consequences of inadvertently doing something like:

31783 *rm -r .\**

31784 Some implementations do not permit the removal of the last link to an executable binary file that is being executed; see the [EBUSY] error in the *unlink()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001. Thus, the *rm* utility can fail to remove such files.

31787 The *-i* option causes *rm* to prompt and read the standard input even if the standard input is not a terminal, but in the absence of *-i* the mode prompting is not done when the standard input is not a terminal.

**31790 EXAMPLES**

31791 1. The following command:

31792 *rm a.out core*

31793 removes the directory entries: **a.out** and **core**.

31794 2. The following command:

31795 *rm -Rf junk*

31796 removes the directory **junk** and all its contents, without prompting.

**31797 RATIONALE**

31798 For absolute clarity, paragraphs (2b) and (3) in the DESCRIPTION of *rm* describing the behavior  
31799 when prompting for confirmation, should be interpreted in the following manner:

31800 *if ((NOT f\_option) AND  
(not\_writable AND input\_is\_terminal) OR i\_option))*

31802 The exact format of the interactive prompts is unspecified. Only the general nature of the  
31803 contents of prompts are specified because implementations may desire more descriptive  
31804 prompts than those used on historical implementations. Therefore, an application not using the  
31805 **-f** option, or using the **-i** option, relies on the system to provide the most suitable dialog directly  
31806 with the user, based on the behavior specified.

31807 The **-r** option is historical practice on all known systems. The synonym **-R** option is provided  
31808 for consistency with the other utilities in this volume of IEEE Std 1003.1-2001 that provide  
31809 options requesting recursive descent through the file hierarchy.

31810 The behavior of the **-f** option in historical versions of *rm* is inconsistent. In general, along with  
31811 “forcing” the unlink without prompting for permission, it always causes diagnostic messages to  
31812 be suppressed and the exit status to be unmodified for nonexistent operands and files that  
31813 cannot be unlinked. In some versions, however, the **-f** option suppresses usage messages and  
31814 system errors as well. Suppressing such messages is not a service to either shell scripts or users.

31815 It is less clear that error messages regarding files that cannot be unlinked (removed) should be  
31816 suppressed. Although this is historical practice, this volume of IEEE Std 1003.1-2001 does not  
31817 permit the **-f** option to suppress such messages.

31818 When given the **-r** and **-i** options, historical versions of *rm* prompt the user twice for each  
31819 directory, once before removing its contents and once before actually attempting to delete the  
31820 directory entry that names it. This allows the user to “prune” the file hierarchy walk. Historical  
31821 versions of *rm* were inconsistent in that some did not do the former prompt for directories  
31822 named on the command line and others had obscure prompting behavior when the **-i** option  
31823 was specified and the permissions of the file did not permit writing. The POSIX Shell and  
31824 Utilities *rm* differs little from historic practice, but does require that prompts be consistent.  
31825 Historical versions of *rm* were also inconsistent in that prompts were done to both standard  
31826 output and standard error. This volume of IEEE Std 1003.1-2001 requires that prompts be done  
31827 to standard error, for consistency with *cp* and *mv*, and to allow historical extensions to *rm* that  
31828 provide an option to list deleted files on standard output.

31829 The *rm* utility is required to descend to arbitrary depths so that any file hierarchy may be  
31830 deleted. This means, for example, that the *rm* utility cannot run out of file descriptors during its  
31831 descent (that is, if the number of file descriptors is limited, *rm* cannot be implemented in the  
31832 historical fashion where one file descriptor is used per directory level). Also, *rm* is not permitted  
31833 to fail because of path length restrictions, unless an operand specified by the user is longer than  
31834 {PATH\_MAX}.

31835 The *rm* utility removes symbolic links themselves, not the files they refer to, as a consequence of  
31836 the dependence on the *unlink()* functionality, per the DESCRIPTION. When removing  
31837 hierarchies with **-r** or **-R**, the prohibition on following symbolic links has to be made explicit.

### 31838 FUTURE DIRECTIONS

31839 None.

### 31840 SEE ALSO

31841 *rmdir*, the System Interfaces volume of IEEE Std 1003.1-2001, *remove()*, *rmdir()*, *unlink()*

### 31842 CHANGE HISTORY

31843 First released in Issue 2.

### 31844 Issue 5

31845 The FUTURE DIRECTIONS section is added.

**31846 Issue 6**

31847       Text is added to clarify actions relating to symbolic links as specified in the IEEE P1003.2b draft  
31848       standard.

**31849 NAME**

31850        **rmdel** — remove a delta from an SCCS file (**DEVELOPMENT**)

**31851 SYNOPSIS**

31852 XSI        **rmdel -r SID file...**

31853

**31854 DESCRIPTION**

31855        The **rmdel** utility shall remove the delta specified by the SID from each named SCCS file. The  
31856        delta to be removed shall be the most recent delta in its branch in the delta chain of each named  
31857        SCCS file. In addition, the application shall ensure that the SID specified is not that of a version  
31858        being edited for the purpose of making a delta; that is, if a *p-file* (see *get*) exists for the named  
31859        SCCS file, the SID specified shall not appear in any entry of the *p-file*.

31860        Removal of a delta shall be restricted to:

- 31861        1. The user who made the delta
- 31862        2. The owner of the SCCS file
- 31863        3. The owner of the directory containing the SCCS file

**31864 OPTIONS**

31865        The **rmdel** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
31866        12.2, Utility Syntax Guidelines.

31867        The following option shall be supported:

31868        **-r SID**        Specify the SCCS identification string (*SID*) of the delta to be deleted.

**31869 OPERANDS**

31870        The following operand shall be supported:

31871        *file*        A pathname of an existing SCCS file or a directory. If *file* is a directory, the **rmdel**  
31872        utility shall behave as though each file in the directory were specified as a named  
31873        file, except that non-SCCS files (last component of the pathname does not begin  
31874        with s.) and unreadable files shall be silently ignored.

31875        If exactly one *file* operand appears, and it is ‘-’, the standard input shall be read;  
31876        each line of the standard input is taken to be the name of an SCCS file to be  
31877        processed. Non-SCCS files and unreadable files shall be silently ignored.

**31878 STDIN**

31879        The standard input shall be a text file used only when the *file* operand is specified as ‘-’. Each  
31880        line of the text file shall be interpreted as an SCCS pathname.

**31881 INPUT FILES**

31882        The SCCS files shall be files of unspecified format.

**31883 ENVIRONMENT VARIABLES**

31884        The following environment variables shall affect the execution of **rmdel**:

31885        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
31886        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31887        Internationalization Variables for the precedence of internationalization variables  
31888        used to determine the values of locale categories.)

31889        *LC\_ALL*        If set to a non-empty string value, override the values of all the other  
31890        internationalization variables.

|       |                               |                                                                                                                                                                                                                                                                     |
|-------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31891 | <i>LC_CTYPE</i>               | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).                                                                           |
| 31894 | <i>LC_MESSAGES</i>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                        |
| 31897 | <i>NLSPATH</i>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                               |
| 31898 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                     |
| 31899 |                               | Default.                                                                                                                                                                                                                                                            |
| 31900 | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                     |
| 31901 |                               | Not used.                                                                                                                                                                                                                                                           |
| 31902 | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                     |
| 31903 |                               | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                      |
| 31904 | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                     |
| 31905 |                               | The SCCS files shall be files of unspecified format. During processing of a <i>file</i> , a temporary <i>x-file</i> , as described in <i>admin</i> , may be created and deleted; a locking <i>z-file</i> , as described in <i>get</i> , may be created and deleted. |
| 31908 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                     |
| 31909 |                               | None.                                                                                                                                                                                                                                                               |
| 31910 | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                     |
| 31911 |                               | The following exit values shall be returned:                                                                                                                                                                                                                        |
| 31912 |                               | 0 Successful completion.                                                                                                                                                                                                                                            |
| 31913 |                               | >0 An error occurred.                                                                                                                                                                                                                                               |
| 31914 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                     |
| 31915 |                               | Default.                                                                                                                                                                                                                                                            |
| 31916 | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                                                                                     |
| 31917 |                               | None.                                                                                                                                                                                                                                                               |
| 31918 | <b>EXAMPLES</b>               |                                                                                                                                                                                                                                                                     |
| 31919 |                               | None.                                                                                                                                                                                                                                                               |
| 31920 | <b>RATIONALE</b>              |                                                                                                                                                                                                                                                                     |
| 31921 |                               | None.                                                                                                                                                                                                                                                               |
| 31922 | <b>FUTURE DIRECTIONS</b>      |                                                                                                                                                                                                                                                                     |
| 31923 |                               | None.                                                                                                                                                                                                                                                               |
| 31924 | <b>SEE ALSO</b>               |                                                                                                                                                                                                                                                                     |
| 31925 |                               | <i>admin, delta, get, prs</i>                                                                                                                                                                                                                                       |
| 31926 | <b>CHANGE HISTORY</b>         |                                                                                                                                                                                                                                                                     |
| 31927 |                               | First released in Issue 2.                                                                                                                                                                                                                                          |
| 31928 | <b>Issue 6</b>                |                                                                                                                                                                                                                                                                     |
| 31929 |                               | The normative text is reworded to avoid use of the term “must” for application requirements.                                                                                                                                                                        |

**31930 NAME**

31931        **rmdir** — remove directories

**31932 SYNOPSIS**

31933        **rmdir [-p] dir...**

**31934 DESCRIPTION**

31935        The **rmdir** utility shall remove the directory entry specified by each *dir* operand.

31936        For each *dir* operand, the **rmdir** utility shall perform actions equivalent to the **rmdir()** function  
31937        called with the *dir* operand as its only argument.

31938        Directories shall be processed in the order specified. If a directory and a subdirectory of that  
31939        directory are specified in a single invocation of the **rmdir** utility, the application shall specify the  
31940        subdirectory before the parent directory so that the parent directory will be empty when the  
31941        **rmdir** utility tries to remove it.

**31942 OPTIONS**

31943        The **rmdir** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
31944        12.2, Utility Syntax Guidelines.

31945        The following option shall be supported:

31946        **-p**        Remove all directories in a pathname. For each *dir* operand:

31947              1. The directory entry it names shall be removed.

31948              2. If the *dir* operand includes more than one pathname component, effects  
31949              equivalent to the following command shall occur:

31950              **rmdir -p \$(dirname dir)**

**31951 OPERANDS**

31952        The following operand shall be supported:

31953        *dir*        A pathname of an empty directory to be removed.

**31954 STDIN**

31955        Not used.

**31956 INPUT FILES**

31957        None.

**31958 ENVIRONMENT VARIABLES**

31959        The following environment variables shall affect the execution of **rmdir**:

31960        **LANG**        Provide a default value for the internationalization variables that are unset or null.  
31961              (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
31962              Internationalization Variables for the precedence of internationalization variables  
31963              used to determine the values of locale categories.)

31964        **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
31965              internationalization variables.

31966        **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
31967              characters (for example, single-byte as opposed to multi-byte characters in  
31968              arguments).

31969        **LC\_MESSAGES**

31970              Determine the locale that should be used to affect the format and contents of  
31971              diagnostic messages written to standard error.

31972 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

31973 **ASYNCHRONOUS EVENTS**

31974 Default.

31975 **STDOUT**

31976 Not used.

31977 **STDERR**

31978 The standard error shall be used only for diagnostic messages.

31979 **OUTPUT FILES**

31980 None.

31981 **EXTENDED DESCRIPTION**

31982 None.

31983 **EXIT STATUS**

31984 The following exit values shall be returned:

31985 0 Each directory entry specified by a *dir* operand was removed successfully.

31986 >0 An error occurred.

31987 **CONSEQUENCES OF ERRORS**

31988 Default.

31989 **APPLICATION USAGE**

31990 The definition of an empty directory is one that contains, at most, directory entries for dot and  
31991 dot-dot.

31992 **EXAMPLES**

31993 If a directory **a** in the current directory is empty except it contains a directory **b** and **a/b** is empty  
31994 except it contains a directory **c**:

31995 `rmdir -p a/b/c`

31996 removes all three directories.

31997 **RATIONALE**

31998 On historical System V systems, the **-p** option also caused a message to be written to the  
31999 standard output. The message indicated whether the whole path was removed or whether part  
32000 of the path remained for some reason. The **STDERR** section requires this diagnostic when the  
32001 entire path specified by a *dir* operand is not removed, but does not allow the status message  
32002 reporting success to be written as a diagnostic.

32003 The *rmdir* utility on System V also included a **-s** option that suppressed the informational  
32004 message output by the **-p** option. This option has been omitted because the informational  
32005 message is not specified by this volume of IEEE Std 1003.1-2001.

32006 **FUTURE DIRECTIONS**

32007 None.

32008 **SEE ALSO**

32009 *rm*, the System Interfaces volume of IEEE Std 1003.1-2001, *remove()*, *rmdir()*, *unlink()*

32010 **CHANGE HISTORY**

32011 First released in Issue 2.

**32012 Issue 6**

32013 The normative text is reworded to avoid use of the term “must” for application requirements.

## 32014 NAME

32015 sact — print current SCCS file-editing activity (**DEVELOPMENT**)

## 32016 SYNOPSIS

32017 XSI sact *file...*

32018

## 32019 DESCRIPTION

32020 The *sact* utility shall inform the user of any impending deltas to a named SCCS file by writing a  
32021 list to standard output. This situation occurs when *get -e* has been executed previously without  
32022 a subsequent execution of *delta*, *unget*, or *sccs unedit*.

## 32023 OPTIONS

32024 None.

## 32025 OPERANDS

32026 The following operand shall be supported:

32027 *file* A pathname of an existing SCCS file or a directory. If *file* is a directory, the *sact*  
32028 utility shall behave as though each file in the directory were specified as a named  
32029 file, except that non-SCCS files (last component of the pathname does not begin  
32030 with **s**.) and unreadable files shall be silently ignored.

32031 If exactly one *file* operand appears, and it is '**-**', the standard input shall be read;  
32032 each line of the standard input shall be taken to be the name of an SCCS file to be  
32033 processed. Non-SCCS files and unreadable files shall be silently ignored.

## 32034 STDIN

32035 The standard input shall be a text file used only when the *file* operand is specified as '**-**'. Each  
32036 line of the text file shall be interpreted as an SCCS pathname.

## 32037 INPUT FILES

32038 Any SCCS files interrogated are files of an unspecified format.

## 32039 ENVIRONMENT VARIABLES

32040 The following environment variables shall affect the execution of *sact*:

32041 *LANG* Provide a default value for the internationalization variables that are unset or null.  
32042 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
32043 Internationalization Variables for the precedence of internationalization variables  
32044 used to determine the values of locale categories.)

32045 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
32046 internationalization variables.

32047 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
32048 characters (for example, single-byte as opposed to multi-byte characters in  
32049 arguments and input files).

32050 *LC\_MESSAGES*

32051 Determine the locale that should be used to affect the format and contents of  
32052 diagnostic messages written to standard error.

32053 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

## 32054 ASYNCHRONOUS EVENTS

32055 Default.

**32056 STDOUT**

32057       The output for each named file shall consist of a line in the following format:

32058        "`%sΔ%Δ%Δ%Δ%Δ\n`", `<SID>`, `<new SID>`, `<login>`, `<date>`, `<time>`

32059        `<SID>`      Specifies the SID of a delta that currently exists in the SCCS file to which changes  
32060        are made to make the new delta.

32061        `<new SID>`   Specifies the SID for the new delta to be created.

32062        `<login>`     Contains the login name of the user who makes the delta (that is, who executed a  
32063        *get* for editing).

32064        `<date>`     Contains the date that *get -e* was executed, in the format used by the *prs :D:* data  
32065        keyword.

32066        `<time>`     Contains the time that *get -e* was executed, in the format used by the *prs :T:* data  
32067        keyword.

32068       If there is more than one named file or if a directory or standard input is named, each pathname  
32069       shall be written before each of the preceding lines:

32070        "`\n%s:\n`", `<pathname>`

**32071 STDERR**

32072       The standard error shall be used only for optional informative messages concerning SCCS files  
32073       with no impending deltas, and for diagnostic messages.

**32074 OUTPUT FILES**

32075       None.

**32076 EXTENDED DESCRIPTION**

32077       None.

**32078 EXIT STATUS**

32079       The following exit values shall be returned:

32080        0   Successful completion.

32081        >0   An error occurred.

**32082 CONSEQUENCES OF ERRORS**

32083       Default.

**32084 APPLICATION USAGE**

32085       None.

**32086 EXAMPLES**

32087       None.

**32088 RATIONALE**

32089       None.

**32090 FUTURE DIRECTIONS**

32091       None.

**32092 SEE ALSO**

32093       *delta, get, sccs, unget*

**32094 CHANGE HISTORY**

32095      First released in Issue 2.

## 32096 NAME

32097 sccs — front end for the SCCS subsystem (DEVELOPMENT)

## 32098 SYNOPSIS

32099 XSI sccs [-r][-d path][-p path] command [options...][operands...]

32100

## 32101 DESCRIPTION

32102 The *sccs* utility is a front end to the SCCS programs. It also includes the capability to run set-  
32103 user-id to another user to provide additional protection.32104 The *sccs* utility shall invoke the specified *command* with the specified *options* and *operands*. By  
32105 default, each of the *operands* shall be modified by prefixing it with the string "SCCS/s .".32106 The *command* can be the name of one of the SCCS utilities in this volume of IEEE Std 1003.1-2001  
32107 (*admin*, *delta*, *get*, *prs*, *rmdel*, *sact*, *unget*, *val*, or *what*) or one of the pseudo-utilities listed in the  
32108 EXTENDED DESCRIPTION section.

## 32109 OPTIONS

32110 The *sccs* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
32111 12.2, Utility Syntax Guidelines, except that *options* operands are actually options to be passed to  
32112 the utility named by *command*. When the portion of the command:32113 *command* [*options* ...] [*operands* ...]32114 is considered, all of the pseudo-utilities used as *command* shall support the Utility Syntax  
32115 Guidelines. Any of the other SCCS utilities that can be invoked in this manner support the  
32116 Guidelines to the extent indicated by their individual OPTIONS sections.32117 The following options shall be supported preceding the *command* operand:32118 **-d path** A pathname of a directory to be used as a root directory for the SCCS files. The  
32119 default shall be the current directory. The **-d** option shall take precedence over the  
32120 *PROJECTDIR* variable. See **-p**.32121 **-p path** A pathname of a directory in which the SCCS files are located. The default shall be  
32122 the **SCCS** directory.32123 The **-p** option differs from the **-d** option in that the **-d** option-argument shall be  
32124 prefixed to the entire pathname and the **-p** option-argument shall be inserted  
32125 before the final component of the pathname. For example:

32126 sccs -d /x -p y get a/b

32127 converts to:

32128 get /x/a/y/s.b

32129 This allows the creation of aliases such as:

32130 alias sysscscs="sccs -d /usr/src"

32131 which is used as:

32132 sysscscs get cmd/who.c

32133 **-r** Invoke *command* with the real user ID of the process, not any effective user ID that  
32134 the *sccs* utility is set to. Certain commands (*admin*, *check*, *clean*, *diffs*, *info*, *rmdel*,  
32135 and *tell*) cannot be run set-user-ID by all users, since this would allow anyone to  
32136 change the authorizations. These commands are always run as the real user.

32137 **OPERANDS**

32138       The following operands shall be supported:

32139       *command*     An SCCS utility name or the name of one of the pseudo-utilities listed in the  
32140                   EXTENDED DESCRIPTION section.

32141       *options*     An option or option-argument to be passed to *command*.

32142       *operands*    An operand to be passed to *command*.

32143 **STDIN**

32144       See the utility description for the specified *command*.

32145 **INPUT FILES**

32146       See the utility description for the specified *command*.

32147 **ENVIRONMENT VARIABLES**

32148       The following environment variables shall affect the execution of *sccs*:

32149       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
32150                   (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
32151                   Internationalization Variables for the precedence of internationalization variables  
32152                   used to determine the values of locale categories.)

32153       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
32154                   internationalization variables.

32155       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
32156                   characters (for example, single-byte as opposed to multi-byte characters in  
32157                   arguments and input files).

32158       *LC\_MESSAGES*

32159       Determine the locale that should be used to affect the format and contents of  
32160                   diagnostic messages written to standard error.

32161       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

32162       *PROJECTDIR*

32163       Provide a default value for the **-d** *path* option. If the value of *PROJECTDIR* begins  
32164                   with a slash, it shall be considered an absolute pathname; otherwise, the value of  
32165                   *PROJECTDIR* is treated as a user name and that user's initial working directory  
32166                   shall be examined for a subdirectory **src** or **source**. If such a directory is found, it  
32167                   shall be used. Otherwise, the value shall be used as a relative pathname.

32168       Additional environment variable effects may be found in the utility description for the specified  
32169                   *command*.

32170 **ASYNCHRONOUS EVENTS**

32171       Default.

32172 **STDOUT**

32173       See the utility description for the specified *command*.

32174 **STDERR**

32175       See the utility description for the specified *command*.

32176 **OUTPUT FILES**

32177       See the utility description for the specified *command*.

## 32178 EXTENDED DESCRIPTION

32179 The following pseudo-utilities shall be supported as *command* operands. All options referred to  
 32180 in the following list are values given in the *options* operands following *command*.

32181 **check** Equivalent to **info**, except that nothing shall be printed if nothing is being edited, and a  
 32182 non-zero exit status shall be returned if anything is being edited. The intent is to have  
 32183 this included in an “install” entry in a makefile to ensure that everything is included  
 32184 into the SCCS file before a version is installed.

32185 **clean** Remove everything from the current directory that can be recreated from SCCS files,  
 32186 but do not remove any files being edited. If the **-b** option is given, branches shall be  
 32187 ignored in the determination of whether they are being edited; this is dangerous if  
 32188 branches are kept in the same directory.

32189 **create** Create an SCCS file, taking the initial contents from the file of the same name. Any  
 32190 options to *admin* are accepted. If the creation is successful, the original files shall be  
 32191 renamed by prefixing the basenames with a comma. These renamed files should be  
 32192 removed after it has been verified that the SCCS files have been created successfully.

32193 **delget** Perform a *delta* on the named files and then *get* new versions. The new versions shall  
 32194 have ID keywords expanded and shall not be editable. Any **-m**, **-p**, **-r**, **-s**, and **-y**  
 32195 options shall be passed to *delta*, and any **-b**, **-c**, **-e**, **-i**, **-k**, **-l**, **-s**, and **-x** options shall be  
 32196 passed to *get*.

32197 **deledit** Equivalent to **delget**, except that the *get* phase shall include the **-e** option. This option  
 32198 is useful for making a checkpoint of the current editing phase. The same options shall  
 32199 be passed to *delta* as described above, and all the options listed for *get* above except **-e**  
 32200 shall be passed to **edit**.

32201 **diffs** Write a difference listing between the current version of the files checked out for  
 32202 editing and the versions in SCCS format. Any **-r**, **-c**, **-i**, **-x**, and **-t** options shall be  
 32203 passed to *get*; any **-l**, **-s**, **-e**, **-f**, **-h**, and **-b** options shall be passed to *diff*. A **-C** option  
 32204 shall be passed to *diff* as **-c**.

32205 **edit** Equivalent to *get -e*.

32206 **fix** Remove the named delta, but leave a copy of the delta with the changes that were in it.  
 32207 It is useful for fixing small compiler bugs, and so on. The application shall ensure that it  
 32208 is followed by a **-r SID** option. Since **fix** does not leave audit trails, it should be used  
 32209 carefully.

32210 **info** Write a listing of all files being edited. If the **-b** option is given, branches (that is, SIDs  
 32211 with two or fewer components) shall be ignored. If a **-u user** option is given, then only  
 32212 files being edited by the named user shall be listed. A **-U** option shall be equivalent to  
 32213 **-u<current user>**.

32214 **print** Write out verbose information about the named files, equivalent to *sccs prs*.

32215 **tell** Write a <newline>-separated list of the files being edited to standard output. Takes the  
 32216 **-b**, **-u**, and **-U** options like **info** and **check**.

32217 **unedit** This is the opposite of an **edit** or a *get -e*. It should be used with caution, since any  
 32218 changes made since the *get* are lost.

## 32219 EXIT STATUS

32220 The following exit values shall be returned:

32221 0 Successful completion.

32222 >0 An error occurred.

### 32223 CONSEQUENCES OF ERRORS

32224 Default.

### 32225 APPLICATION USAGE

32226 Many of the SCCS utilities take directory names as operands as well as specific filenames. The  
32227 pseudo-utilities supported by sccs are not described as having this capability, but are not  
32228 prohibited from doing so.

### 32229 EXAMPLES

32230 1. To get a file for editing, edit it and produce a new delta:

```
32231 sccs get -e file.c
32232 ex file.c
32233 sccs delta file.c
```

32234 2. To get a file from another directory:

```
32235 sccs -p /usr/src/sccs/s. get cc.c
```

32236 or:

```
32237 sccs get /usr/src/sccs/s.cc.c
```

32238 3. To make a delta of a large number of files in the current directory:

```
32239 sccs delta *.c
```

32240 4. To get a list of files being edited that are not on branches:

```
32241 sccs info -b
```

32242 5. To delta everything being edited by the current user:

```
32243 sccs delta $(sccs tell -U)
```

32244 6. In a makefile, to get source files from an SCCS file if it does not already exist:

```
32245 SRCS = <list of source files>
32246 $(SRCS):
32247 sccs get $(REL) $@
```

### 32248 RATIONALE

32249 SCCS and its associated utilities are part of the XSI Development Utilities option within the XSI  
32250 extension.

32251 SCCS is an abbreviation for Source Code Control System. It is a maintenance and enhancement  
32252 tracking tool. When a file is put under SCCS, the source code control system maintains the file  
32253 and, when changes are made, identifies and stores them in the file with the original source code  
32254 and/or documentation. As other changes are made, they too are identified and retained in the  
32255 file.

32256 Retrieval of the original and any set of changes is possible. Any version of the file as it develops  
32257 can be reconstructed for inspection or additional modification. History data can be stored with  
32258 each version, documenting why the changes were made, who made them, and when they were  
32259 made.

**32260 FUTURE DIRECTIONS**

32261        None.

**32262 SEE ALSO**

32263        *admin, delta, get, make, prs, rmdel, sact, unget, val, what*

**32264 CHANGE HISTORY**

32265        First released in Issue 4.

**32266 Issue 6**

32267        In the ENVIRONMENT VARIABLES section, the *PROJECTDIR* description is updated from  
32268        “otherwise, the home directory of a user of that name is examined” to “otherwise, the value of  
32269        *PROJECTDIR* is treated as a user name and that user’s initial working directory is examined”.

32270        The normative text is reworded to avoid use of the term “must” for application requirements.

## 32271 NAME

32272 sed — stream editor

## 32273 SYNOPSIS

32274 sed [-n] *script[file...]*

32275 sed [-n][-e *script*]...[-f *script\_file*]...[*file...*]

## 32276 DESCRIPTION

32277 The *sed* utility is a stream editor that shall read one or more text files, make editing changes according to a script of editing commands, and write the results to standard output. The script shall be obtained from either the *script* operand string or a combination of the option-arguments from the **-e** *script* and **-f** *script\_file* options.

## 32281 OPTIONS

32282 The *sed* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines, except that the order of presentation of the **-e** and **-f** options is significant.

32285 The following options shall be supported:

32286 **-e** *script* Add the editing commands specified by the *script* option-argument to the end of the script of editing commands. The *script* option-argument shall have the same properties as the *script* operand, described in the OPERANDS section.

32289 **-f** *script\_file* Add the editing commands in the file *script\_file* to the end of the script.

32290 **-n** Suppress the default output (in which each line, after it is examined for editing, is written to standard output). Only lines explicitly selected for output are written.

32292 Multiple **-e** and **-f** options may be specified. All commands shall be added to the script in the order specified, regardless of their origin.

## 32294 OPERANDS

32295 The following operands shall be supported:

32296 *file* A pathname of a file whose contents are read and edited. If multiple *file* operands are specified, the named files shall be read in the order specified and the concatenation shall be edited. If no *file* operands are specified, the standard input shall be used.

32300 *script* A string to be used as the script of editing commands. The application shall not present a *script* that violates the restrictions of a text file except that the final character need not be a <newline>.

## 32303 STDIN

32304 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES section.

## 32306 INPUT FILES

32307 The input files shall be text files. The *script\_files* named by the **-f** option shall consist of editing commands.

## 32309 ENVIRONMENT VARIABLES

32310 The following environment variables shall affect the execution of *sed*:

32311 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

|           |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32315     | <i>LC_ALL</i>               | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 32317     | <i>LC_COLLATE</i>           | Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 32320     | <i>LC_CTYPE</i>             | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), and the behavior of character classes within regular expressions.                                                                                                                                                                                                                                                                                                                                                               |
| 32324     | <i>LC_MESSAGES</i>          | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 32327 XSI | <i>NLSPATH</i>              | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 32328     | <b>ASYNCHRONOUS EVENTS</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32329     |                             | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 32330     | <b>STDOUT</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32331     |                             | The input files shall be written to standard output, with the editing commands specified in the script applied. If the <b>-n</b> option is specified, only those input lines selected by the script shall be written to standard output.                                                                                                                                                                                                                                                                                                                                                                                  |
| 32334     | <b>STDERR</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32335     |                             | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 32336     | <b>OUTPUT FILES</b>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32337     |                             | The output files shall be text files whose formats are dependent on the editing commands given.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32338     | <b>EXTENDED DESCRIPTION</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32339     |                             | The <i>script</i> shall consist of editing commands of the following form:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32340     |                             | <i>[address[ , address]]function</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 32341     |                             | where <i>function</i> represents a single-character command verb from the list in <b>Editing Commands in sed</b> (on page 840), followed by any applicable arguments.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 32343     |                             | The command can be preceded by <blank>s and/or semicolons. The function can be preceded by <blank>s. These optional characters shall have no effect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 32345     |                             | In default operation, <i>sed</i> cyclically shall append a line of input, less its terminating <newline>, into the pattern space. Normally the pattern space will be empty, unless a <b>D</b> command terminated the last cycle. The <i>sed</i> utility shall then apply in sequence all commands whose addresses select that pattern space, and at the end of the script copy the pattern space to standard output (except when <b>-n</b> is specified) and delete the pattern space. Whenever the pattern space is written to standard output or a named file, <i>sed</i> shall immediately follow it with a <newline>. |
| 32352     |                             | Some of the editing commands use a hold space to save all or part of the pattern space for subsequent retrieval. The pattern and hold spaces shall each be able to hold at least 8 192 bytes.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 32353     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

- 32354       **Addresses in sed**
- 32355       An address is either a decimal number that counts input lines cumulatively across files, a '\$' character that addresses the last line of input, or a context address (which consists of a BRE, as described in **Regular Expressions in sed**, preceded and followed by a delimiter, usually a slash).
- 32358       An editing command with no addresses shall select every pattern space.
- 32359       An editing command with one address shall select each pattern space that matches the address.
- 32360       An editing command with two addresses shall select the inclusive range from the first pattern space that matches the first address through the next pattern space that matches the second. (If the second address is a number less than or equal to the line number first selected, only one line shall be selected.) Starting at the first line following the selected range, *sed* shall look again for the first address. Thereafter, the process shall be repeated. Omitting either or both of the address components in the following form produces undefined results:
- 32366       `[address[, address]]`
- 32367       **Regular Expressions in sed**
- 32368       The *sed* utility shall support the BREs described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions, with the following additions:
- In a context address, the construction "\cBREc", where c is any character other than backslash or <newline>, shall be identical to "/BRE/". If the character designated by c appears following a backslash, then it shall be considered to be that literal character, which shall not terminate the BRE. For example, in the context address "\xabc\xdefx", the second x stands for itself, so that the BRE is "abcxdef".
  - The escape sequence '\n' shall match a <newline> embedded in the pattern space. A literal <newline> shall not be used in the BRE of a context address or in the substitute function.
  - If an RE is empty (that is, no pattern is specified) *sed* shall behave as if the last RE used in the last command applied (either as an address or as part of a substitute command) was specified.
- 32380       **Editing Commands in sed**
- 32381       In the following list of editing commands, the maximum number of permissible addresses for each function is indicated by [0addr], [1addr], or [2addr], representing zero, one, or two addresses.
- 32384       The argument *text* shall consist of one or more lines. Each embedded <newline> in the text shall be preceded by a backslash. Other backslashes in text shall be removed, and the following character shall be treated literally.
- 32387       The r and w command verbs, and the w flag to the s command, take an optional *rfile* (or *wfile*) parameter, separated from the command verb letter or flag by one or more <blank>s; implementations may allow zero separation as an extension.
- 32390       The argument *rfile* or the argument *wfile* shall terminate the editing command. Each *wfile* shall be created before processing begins. Implementations shall support at least ten *wfile* arguments in the script; the actual number (greater than or equal to 10) that is supported by the implementation is unspecified. The use of the *wfile* parameter shall cause that file to be initially created, if it does not exist, or shall replace the contents of an existing file.
- 32395       The b, r, s, t, w, y, and : command verbs shall accept additional arguments. The following synopses indicate which arguments shall be separated from the command verbs by a single

32397        <space>.

32398        The **a** and **r** commands schedule text for later output. The text specified for the **a** command, and  
 32399        the contents of the file specified for the **r** command, shall be written to standard output just  
 32400        before the next attempt to fetch a line of input when executing the **N** or **n** commands, or when  
 32401        reaching the end of the script. If written when reaching the end of the script, and the **-n** option  
 32402        was not specified, the text shall be written after copying the pattern space to standard output.  
 32403        The contents of the file specified for the **r** command shall be as of the time the output is written,  
 32404        not the time the **r** command is applied. The text shall be output in the order in which the **a** and **r**  
 32405        commands were applied to the input.

32406        Command verbs other than **{**, **a**, **b**, **c**, **i**, **r**, **t**, **w**, **:**, and **#** can be followed by a semicolon, optional  
 32407        <blank>s, and another command verb. However, when the **s** command verb is used with the **w**  
 32408        flag, following it with another command in this manner produces undefined results.

32409        A function can be preceded by one or more '!' characters, in which case the function shall be  
 32410        applied if the addresses do not select the pattern space. Zero or more <blank>s shall be accepted  
 32411        before the first '!' character. It is unspecified whether <blank>s can follow a '!' character, and  
 32412        conforming applications shall not follow a '!' character with <blank>s.

32413        **[2addr] {function**  
 32414        **function**  
 32415        ...  
 32416        **}**        Execute a list of *sed* functions only when the pattern space is selected. The list of  
 32417        *sed* functions shall be surrounded by braces and separated by <newline>s, and  
 32418        conform to the following rules. The braces can be preceded or followed by  
 32419        <blank>s. The functions can be preceded by <blank>s, but shall not be followed  
 32420        by <blank>s. The <right-brace> shall be preceded by a <newline> and can be  
 32421        preceded or followed by <blank>s.

32422        **[1addr]a\**  
 32423        **text**        Write text to standard output as described previously.

32424        **[2addr]b [label]**  
 32425        Branch to the : function bearing the *label*. If *label* is not specified, branch to the end  
 32426        of the script. The implementation shall support *labels* recognized as unique up to  
 32427        at least 8 characters; the actual length (greater than or equal to 8) that shall be  
 32428        supported by the implementation is unspecified. It is unspecified whether  
 32429        exceeding a label length causes an error or a silent truncation.

32430        **[2addr]c\**  
 32431        **text**        Delete the pattern space. With a 0 or 1 address or at the end of a 2-address range,  
 32432        place *text* on the output and start the next cycle.

32433        **[2addr]d**      Delete the pattern space and start the next cycle.

32434        **[2addr]D**      Delete the initial segment of the pattern space through the first <newline> and  
 32435        start the next cycle.

32436        **[2addr]g**      Replace the contents of the pattern space by the contents of the hold space.

32437        **[2addr]G**      Append to the pattern space a <newline> followed by the contents of the hold  
 32438        space.

32439        **[2addr]h**      Replace the contents of the hold space with the contents of the pattern space.

32440        **[2addr]H**      Append to the hold space a <newline> followed by the contents of the pattern  
 32441        space.

|       |                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32442 | <b>[1addr]i\text</b>                  | Write <i>text</i> to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 32444 | <b>[2addr]l</b>                       | (The letter ell.) Write the pattern space to standard output in a visually unambiguous form. The characters listed in the Base Definitions volume of IEEE Std 1003.1-2001, Table 5-1, Escape Sequences and Associated Actions ('\\', '\\a', '\\b', '\\f', '\\r', '\\t', '\\v') shall be written as the corresponding escape sequence; the '\\n' in that table is not applicable. Non-printable characters not in that table shall be written as one three-digit octal number (with a preceding backslash) for each byte in the character (most significant byte first). If the size of a byte on the system is greater than 9 bits, the format used for non-printable characters is implementation-defined. |
| 32453 |                                       | Long lines shall be folded, with the point of folding indicated by writing a backslash followed by a <newline>; the length at which folding occurs is unspecified, but should be appropriate for the output device. The end of each line shall be marked with a '\$'.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 32457 | <b>[2addr]n</b>                       | Write the pattern space to standard output if the default output has not been suppressed, and replace the pattern space with the next line of input, less its terminating <newline>.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 32460 |                                       | If no next line of input is available, the <b>n</b> command verb shall branch to the end of the script and quit without starting a new cycle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 32462 | <b>[2addr]N</b>                       | Append the next line of input, less its terminating <newline>, to the pattern space, using an embedded <newline> to separate the appended material from the original material. Note that the current line number changes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 32465 |                                       | If no next line of input is available, the <b>N</b> command verb shall branch to the end of the script and quit without starting a new cycle or copying the pattern space to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 32468 | <b>[2addr]p</b>                       | Write the pattern space to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 32469 | <b>[2addr]P</b>                       | Write the pattern space, up to the first <newline>, to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 32470 | <b>[1addr]q</b>                       | Branch to the end of the script and quit without starting a new cycle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 32471 | <b>[1addr]r rfile</b>                 | Copy the contents of <i>rfile</i> to standard output as described previously. If <i>rfile</i> does not exist or cannot be read, it shall be treated as if it were an empty file, causing no error condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32474 | <b>[2addr]s/BRE/replacement/flags</b> | Substitute the replacement string for instances of the BRE in the pattern space. Any character other than backslash or <newline> can be used instead of a slash to delimit the BRE and the replacement. Within the BRE and the replacement, the BRE delimiter itself can be used as a literal character if it is preceded by a backslash.                                                                                                                                                                                                                                                                                                                                                                   |
| 32480 |                                       | The replacement string shall be scanned from beginning to end. An ampersand ('&') appearing in the replacement shall be replaced by the string matching the BRE. The special meaning of '&' in this context can be suppressed by preceding it by a backslash. The characters "\n", where <i>n</i> is a digit, shall be replaced by the text matched by the corresponding backreference expression. The special meaning of "\n" where <i>n</i> is a digit in this context, can be suppressed by preceding it by a backslash. For each other backslash ('\') encountered, the following character shall lose its special meaning (if any). The meaning of a '\' immediately followed                          |

32488 by any character other than '&', '\', a digit, or the delimiter character used for  
32489 this command, is unspecified.

32490 A line can be split by substituting a <newline> into it. The application shall escape  
32491 the <newline> in the replacement by preceding it by a backslash. A substitution  
32492 shall be considered to have been performed even if the replacement string is  
32493 identical to the string that it replaces. Any backslash used to alter the default  
32494 meaning of a subsequent character shall be discarded from the BRE or the  
32495 replacement before evaluating the BRE or using the replacement.

32496 The value of *flags* shall be zero or more of:

32497 **n** Substitute for the *n*th occurrence only of the BRE found within the  
32498 pattern space.  
32499 **g** Globally substitute for all non-overlapping instances of the BRE  
32500 rather than just the first one. If both **g** and **n** are specified, the results  
32501 are unspecified.  
32502 **p** Write the pattern space to standard output if a replacement was  
32503 made.  
32504 **w** *wfile* Write. Append the pattern space to *wfile* if a replacement was made.  
32505 A conforming application shall precede the *wfile* argument with one  
32506 or more <blank>s. If the **w** flag is not the last flag value given in a  
32507 concatenation of multiple flag values, the results are undefined.

32508 **[2addr]t [label]**

32509 Test. Branch to the : command verb bearing the *label* if any substitutions have been  
32510 made since the most recent reading of an input line or execution of a **t**. If *label* is  
32511 not specified, branch to the end of the script.

32512 **[2addr]w wfile**

32513 Append (write) the pattern space to *wfile*.

32514 **[2addr]x** Exchange the contents of the pattern and hold spaces.

32515 **[2addr]y/string1/string2/**

32516 Replace all occurrences of characters in *string1* with the corresponding characters  
32517 in *string2*. If a backslash followed by an 'n' appear in *string1* or *string2*, the two  
32518 characters shall be handled as a single <newline>. If the number of characters in  
32519 *string1* and *string2* are not equal, or if any of the characters in *string1* appear more  
32520 than once, the results are undefined. Any character other than backslash or  
32521 <newline> can be used instead of slash to delimit the strings. If the delimiter is not  
32522 *n*, within *string1* and *string2*, the delimiter itself can be used as a literal character if  
32523 it is preceded by a backslash. If a backslash character is immediately followed by a  
32524 backslash character in *string1* or *string2*, the two backslash characters shall be  
32525 counted as a single literal backslash character. The meaning of a backslash  
32526 followed by any character that is not 'n', a backslash, or the delimiter character is  
32527 undefined.

32528 **[0addr]:label** Do nothing. This command bears a *label* to which the **b** and **t** commands branch.

32529 **[1addr]=** Write the following to standard output:

32530 " %d\n" , <current line number>

32531 **[0addr]** Ignore this empty command.

32532 [0addr]# Ignore the '#' and the remainder of the line (treat them as a comment), with the  
32533 single exception that if the first two characters in the script are "#n", the default  
32534 output shall be suppressed; this shall be the equivalent of specifying -n on the  
32535 command line.

### 32536 EXIT STATUS

32537 The following exit values shall be returned:

- 32538 0 Successful completion.  
32539 >0 An error occurred.

### 32540 CONSEQUENCES OF ERRORS

32541 Default.

### 32542 APPLICATION USAGE

32543 Regular expressions match entire strings, not just individual lines, but a <newline> is matched  
32544 by '\n' in a sed RE; a <newline> is not allowed by the general definition of regular expression in  
32545 IEEE Std 1003.1-2001. Also note that '\n' cannot be used to match a <newline> at the end of an  
32546 arbitrary input line; <newline>s appear in the pattern space as a result of the N editing  
32547 command.

### 32548 EXAMPLES

32549 This sed script simulates the BSD cat -s command, squeezing excess blank lines from standard  
32550 input.

```
32551 sed -n '
32552 # Write non-empty lines.
32553 ./ {
32554 p
32555 d
32556 }
32557 # Write a single empty line, then look for more empty lines.
32558 /^$/ p
32559 # Get next line, discard the held <newline> (empty line),
32560 # and look for more empty lines.
32561 :Empty
32562 /^$/ {
32563 N
32564 s/.//
32565 b Empty
32566 }
32567 # Write the non-empty line before going back to search
32568 # for the first in a set of empty lines.
32569 p
32570 ,
```

### 32571 RATIONALE

32572 This volume of IEEE Std 1003.1-2001 requires implementations to support at least ten distinct  
32573 wfiles, matching historical practice on many implementations. Implementations are encouraged  
32574 to support more, but conforming applications should not exceed this limit.

32575 The exit status codes specified here are different from those in System V. System V returns 2 for  
32576 garbled sed commands, but returns zero with its usage message or if the input file could not be  
32577 opened. The standard developers considered this to be a bug.

32578 The manner in which the **I** command writes non-printable characters was changed to avoid the  
32579 historical backspace-overstrike method, and other requirements to achieve unambiguous output  
32580 were added. See the RATIONALE for *ed* for details of the format chosen, which is the same as  
32581 that chosen for *sed*.

32582 This volume of IEEE Std 1003.1-2001 requires implementations to provide pattern and hold  
32583 spaces of at least 8 192 bytes, larger than the 4 000 bytes spaces used by some historical  
32584 implementations, but less than the 20 480 bytes limit used in an early proposal. Implementations  
32585 are encouraged to allocate dynamically larger pattern and hold spaces as needed.

32586 The requirements for acceptance of <blank>s and <space>s in command lines has been made  
32587 more explicit than in early proposals to describe clearly the historical practice and to remove  
32588 confusion about the phrase “protect initial blanks [sic] and tabs from the stripping that is done  
32589 on every script line” that appears in much of the historical documentation of the *sed* utility  
32590 description of text. (Not all implementations are known to have stripped <blank>s from text  
32591 lines, although they all have allowed leading <blank>s preceding the address on a command  
32592 line.)

32593 The treatment of ‘#’ comments differs from the SVID which only allows a comment as the first  
32594 line of the script, but matches BSD-derived implementations. The comment character is treated  
32595 as a command, and it has the same properties in terms of being accepted with leading <blank>s;  
32596 the BSD implementation has historically supported this.

32597 Early proposals required that a *script\_file* have at least one non-comment line. Some historical  
32598 implementations have behaved in unexpected ways if this were not the case. The standard  
32599 developers considered that this was incorrect behavior and that application developers should  
32600 not have to avoid this feature. A correct implementation of this volume of IEEE Std 1003.1-2001  
32601 shall permit *script\_files* that consist only of comment lines.

32602 Early proposals indicated that if **-e** and **-f** options were intermixed, all **-e** options were  
32603 processed before any **-f** options. This has been changed to process them in the order presented  
32604 because it matches historical practice and is more intuitive.

32605 The treatment of the **p** flag to the **s** command differs between System V and BSD-based systems  
32606 when the default output is suppressed. In the two examples:

```
32607 echo a | sed 's/a/A/p'
32608 echo a | sed -n 's/a/A/p'
```

32609 this volume of IEEE Std 1003.1-2001, BSD, System V documentation, and the SVID indicate that  
32610 the first example should write two lines with **A**, whereas the second should write one. Some  
32611 System V systems write the **A** only once in both examples because the **p** flag is ignored if the **-n**  
32612 option is not specified.

32613 This is a case of a diametrical difference between systems that could not be reconciled through  
32614 the compromise of declaring the behavior to be unspecified. The SVID/BSD/System V  
32615 documentation behavior was adopted for this volume of IEEE Std 1003.1-2001 because:

- 32616 • No known documentation for any historic system describes the interaction between the **p**  
32617 flag and the **-n** option.
- 32618 • The selected behavior is more correct as there is no technical justification for any interaction  
32619 between the **p** flag and the **-n** option. A relationship between **-n** and the **p** flag might imply  
32620 that they are only used together, but this ignores valid scripts that interrupt the cyclical  
32621 nature of the processing through the use of the **D**, **d**, **q**, or branching commands. Such scripts  
32622 rely on the **p** suffix to write the pattern space because they do not make use of the default  
32623 output at the “bottom” of the script.

- 32624     • Because the **-n** option makes the **p** flag unnecessary, any interaction would only be useful if  
32625     **sed** scripts were written to run both with and without the **-n** option. This is believed to be  
32626     unlikely. It is even more unlikely that programmers have coded the **p** flag expecting it to be  
32627     unnecessary. Because the interaction was not documented, the likelihood of a programmer  
32628     discovering the interaction and depending on it is further decreased.
- 32629     • Finally, scripts that break under the specified behavior produce too much output instead of  
32630     too little, which is easier to diagnose and correct.

32631     The form of the substitute command that uses the **n** suffix was limited to the first 512 matches in  
32632     an early proposal. This limit has been removed because there is no reason an editor processing  
32633     lines of {LINE\_MAX} length should have this restriction. The command **s/a/A/2047** should be  
32634     able to substitute the 2 047th occurrence of **a** on a line.

32635     The **b**, **t**, and **:** commands are documented to ignore leading white space, but no mention is  
32636     made of trailing white space. Historical implementations of **sed** assigned different locations to  
32637     the labels '**x**' and "**x**". This is not useful, and leads to subtle programming errors, but it is  
32638     historical practice, and changing it could theoretically break working scripts. Implementors are  
32639     encouraged to provide warning messages about labels that are never used or jumps to labels  
32640     that do not exist.

32641     Historically, the **sed !** and **}** editing commands did not permit multiple commands on a single  
32642     line using a semicolon as a command delimiter. Implementations are permitted, but not  
32643     required, to support this extension.

## 32644 FUTURE DIRECTIONS

32645     None.

## 32646 SEE ALSO

32647       *awk, ed, grep*

## 32648 CHANGE HISTORY

32649       First released in Issue 2.

### 32650 Issue 5

32651       The FUTURE DIRECTIONS section is added.

### 32652 Issue 6

32653       The following new requirements on POSIX implementations derive from alignment with the  
32654       Single UNIX Specification:

- Implementations are required to support at least ten *wfile* arguments in an editing command.

32656       The EXTENDED DESCRIPTION is changed to align with the IEEE P1003.2b draft standard.

32657       IEEE PASC Interpretation 1003.2 #190 is applied.

32658       IEEE PASC Interpretation 1003.2 #203 is applied, clarifying the meaning of the backslash escape  
32659       sequences in a replacement string for a BRE.

**32660 NAME**

32661        sh — shell, the standard command language interpreter

**32662 SYNOPSIS**

```
32663 sh [-abCefhimnuvx][+o option][+abCefhimnuvx][+o option]
32664 [command_file [argument...]]
32665 sh -c[-abCefhimnuvx][+o option][+abCefhimnuvx][+o option]command_string
32666 [command_name [argument...]]
32667 sh -s[-abCefhimnuvx][+o option][+abCefhimnuvx][+o option][argument]
```

**32668 DESCRIPTION**

32669        The *sh* utility is a command language interpreter that shall execute commands read from a  
32670        command line string, the standard input, or a specified file. The application shall ensure that the  
32671        commands to be executed are expressed in the language described in Chapter 2 (on page 29).

32672        Pathname expansion shall not fail due to the size of a file.

32673        Shell input and output redirections have an implementation-defined offset maximum that is  
32674        established in the open file description.

**32675 OPTIONS**

32676        The *sh* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
32677        Utility Syntax Guidelines, with an extension for support of a leading plus sign ('+') as noted  
32678        below.

32679        The **-a**, **-b**, **-C**, **-e**, **-f**, **-m**, **-n**, **-o option**, **-u**, **-v**, and **-x** options are described as part of the *set*  
32680        utility in Section 2.14 (on page 64). The option letters derived from the *set* special built-in shall  
32681        also be accepted with a leading plus sign ('+') instead of a leading hyphen (meaning the reverse  
32682        case of the option as described in this volume of IEEE Std 1003.1-2001).

32683        The following additional options shall be supported:

32684        **-c**        Read commands from the *command\_string* operand. Set the value of special  
32685        parameter 0 (see Section 2.5.2 (on page 34)) from the value of the *command\_name*  
32686        operand and the positional parameters (\$1, \$2, and so on) in sequence from the  
32687        remaining *argument* operands. No commands shall be read from the standard  
32688        input.

32689        **-i**        Specify that the shell is *interactive*; see below. An implementation may treat  
32690        specifying the **-i** option as an error if the real user ID of the calling process does  
32691        not equal the effective user ID or if the real group ID does not equal the effective  
32692        group ID.

32693        **-s**        Read commands from the standard input.

32694        If there are no operands and the **-c** option is not specified, the **-s** option shall be assumed.

32695        If the **-i** option is present, or if there are no operands and the shell's standard input and standard  
32696        error are attached to a terminal, the shell is considered to be *interactive*.

**32697 OPERANDS**

32698        The following operands shall be supported:

32699        **-**        A single hyphen shall be treated as the first operand and then ignored. If both '**-**'  
32700        and "**--**" are given as arguments, or if other operands precede the single hyphen,  
32701        the results are undefined.

32702        **argument**    The positional parameters (\$1, \$2, and so on) shall be set to *arguments*, if any.

32703       *command\_file* The pathname of a file containing commands. If the pathname contains one or  
32704        more slash characters, the implementation attempts to read that file; the file need  
32705        not be executable. If the pathname does not contain a slash character:

- 32706        • The implementation shall attempt to read that file from the current working  
32707        directory; the file need not be executable.
- 32708        • If the file is not in the current working directory, the implementation may  
32709        perform a search for an executable file using the value of *PATH*, as described in  
32710        Section 2.9.1.1 (on page 48).

32711        Special parameter 0 (see Section 2.5.2 (on page 34)) shall be set to the value of  
32712        *command\_file*. If *sh* is called using a synopsis form that omits *command\_file*, special  
32713        parameter 0 shall be set to the value of the first argument passed to *sh* from its  
32714        parent (for example, *argv[0]* for a C program), which is normally a pathname used  
32715        to execute the *sh* utility.

32716        *command\_name*  
32717           A string assigned to special parameter 0 when executing the commands in  
32718           *command\_string*. If *command\_name* is not specified, special parameter 0 shall be set  
32719           to the value of the first argument passed to *sh* from its parent (for example, *argv[0]*  
32720           for a C program), which is normally a pathname used to execute the *sh* utility.

32721        *command\_string*  
32722           A string that shall be interpreted by the shell as one or more commands, as if the  
32723           string were the argument to the *system()* function defined in the System Interfaces  
32724           volume of IEEE Std 1003.1-2001. If the *command\_string* operand is an empty string,  
32725           *sh* shall exit with a zero exit status.

## 32726    STDIN

32727       The standard input shall be used only if one of the following is true:

- 32728       • The *-s* option is specified.
- 32729       • The *-c* option is not specified and no operands are specified.
- 32730       • The script executes one or more commands that require input from standard input (such as a  
32731        *read* command that does not redirect its input).

32732       See the INPUT FILES section.

32733       When the shell is using standard input and it invokes a command that also uses standard input,  
32734       the shell shall ensure that the standard input file pointer points directly after the command it has  
32735       read when the command begins execution. It shall not read ahead in such a manner that any  
32736       characters intended to be read by the invoked command are consumed by the shell (whether  
32737       interpreted by the shell or not) or that characters that are not read by the invoked command are  
32738       not seen by the shell. When the command expecting to read standard input is started  
32739       asynchronously by an interactive shell, it is unspecified whether characters are read by the  
32740       command or interpreted by the shell.

32741       If the standard input to *sh* is a FIFO or terminal device and is set to non-blocking reads, then *sh*  
32742       shall enable blocking reads on standard input. This shall remain in effect when the command  
32743       completes.

## 32744    INPUT FILES

32745       The input file shall be a text file, except that line lengths shall be unlimited. If the input file is  
32746       empty or consists solely of blank lines or comments, or both, *sh* shall exit with a zero exit status.

## 32747 ENVIRONMENT VARIABLES

32748 The following environment variables shall affect the execution of *sh*:

|       |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32749 | <i>ENV</i>      | This variable, when and only when an interactive shell is invoked, shall be subjected to parameter expansion (see Section 2.6.2 (on page 37)) by the shell, and the resulting value shall be used as a pathname of a file containing shell commands to execute in the current environment. The file need not be executable. If the expanded value of <i>ENV</i> is not an absolute pathname, the results are unspecified. <i>ENV</i> shall be ignored if the real and effective user IDs or real and effective group IDs of the process are different.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 32756 | <i>FCEDIT</i>   | This variable, when expanded by the shell, shall determine the default value for the <code>-e editor</code> option's <i>editor</i> option-argument. If <i>FCEDIT</i> is null or unset, <i>ed</i> shall be used as the editor. This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 32760 | <i>HISTFILE</i> | Determine a pathname naming a command history file. If the <i>HISTFILE</i> variable is not set, the shell may attempt to access or create a file <code>.sh_history</code> in the directory referred to by the <i>HOME</i> environment variable. If the shell cannot obtain both read and write access to, or create, the history file, it shall use an unspecified mechanism that allows the history to operate properly. (References to history "file" in this section shall be understood to mean this unspecified mechanism in such cases.) An implementation may choose to access this variable only when initializing the history file; this initialization shall occur when <i>fc</i> or <i>sh</i> first attempt to retrieve entries from, or add entries to, the file, as the result of commands issued by the user, the file named by the <i>ENV</i> variable, or implementation-defined system start-up files. Implementations may choose to disable the history list mechanism for users with appropriate privileges who do not set <i>HISTFILE</i> ; the specific circumstances under which this occurs are implementation-defined. If more than one instance of the shell is using the same history file, it is unspecified how updates to the history file from those shells interact. As entries are deleted from the history file, they shall be deleted oldest first. It is unspecified when history file entries are physically removed from the history file. This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option. |
| 32779 | <i>HISTSIZE</i> | Determine a decimal number representing the limit to the number of previous commands that are accessible. If this variable is unset, an unspecified default greater than or equal to 128 shall be used. The maximum number of commands in the history list is unspecified, but shall be at least 128. An implementation may choose to access this variable only when initializing the history file, as described under <i>HISTFILE</i> . Therefore, it is unspecified whether changes made to <i>HISTSIZE</i> after the history file has been initialized are effective.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 32786 | <i>HOME</i>     | Determine the pathname of the user's home directory. The contents of <i>HOME</i> are used in tilde expansion as described in Section 2.6.1 (on page 37). This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 32790 | <i>IFS</i>      | (Input Field Separators.) A string treated as a list of characters that shall be used for field splitting and to split lines into words with the <i>read</i> command. See Section 2.6.5 (on page 42). If <i>IFS</i> is not set, the shell shall behave as if the value of <i>IFS</i> were <code>&lt;space&gt;</code> , <code>&lt;tab&gt;</code> , and <code>&lt;newline&gt;</code> . Implementations may ignore the value of <i>IFS</i> in the environment at the time <i>sh</i> is invoked, treating <i>IFS</i> as if it were not set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

|       |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32795 | <i>LANG</i>               | Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 32799 | <i>LC_ALL</i>             | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32801 | <i>LC_COLLATE</i>         | Determine the behavior of range expressions, equivalence classes, and multi-character collating elements within pattern matching.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 32804 | <i>LC_CTYPE</i>           | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), which characters are defined as letters (character class <b>alpha</b> ), and the behavior of character classes within pattern matching.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 32808 | <i>LC_MESSAGES</i>        | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 32811 | <i>MAIL</i>               | Determine a pathname of the user's mailbox file for purposes of incoming mail notification. If this variable is set, the shell shall inform the user if the file named by the variable is created or if its modification time has changed. Informing the user shall be accomplished by writing a string of unspecified format to standard error prior to the writing of the next primary prompt string. Such check shall be performed only after the completion of the interval defined by the <i>MAILCHECK</i> variable after the last such check. The user shall be informed only if <i>MAIL</i> is set and <i>MAILPATH</i> is not set. This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option. |
| 32820 | <i>MAILCHECK</i>          | Establish a decimal integer value that specifies how often (in seconds) the shell shall check for the arrival of mail in the files specified by the <i>MAILPATH</i> or <i>MAIL</i> variables. The default value shall be 600 seconds. If set to zero, the shell shall check before issuing each primary prompt. This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option.                                                                                                                                                                                                                                                                                                                           |
| 32827 | <i>MAILPATH</i>           | Provide a list of pathnames and optional messages separated by colons. If this variable is set, the shell shall inform the user if any of the files named by the variable are created or if any of their modification times change. (See the preceding entry for <i>MAIL</i> for descriptions of mail arrival and user informing.) Each pathname can be followed by '%' and a string that shall be subjected to parameter expansion and written to standard error when the modification time changes. If a '%' character in the pathname is preceded by a backslash, it shall be treated as a literal '%' in the pathname. The default message is unspecified.                                                                                                                          |
| 32835 |                           | The <i>MAILPATH</i> environment variable takes precedence over the <i>MAIL</i> variable. This volume of IEEE Std 1003.1-2001 specifies the effects of this variable only for systems supporting the User Portability Utilities option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 32838 | <i>XSI</i> <i>NLSPATH</i> | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 32839 | <i>PATH</i>               | Establish a string formatted as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables, used to effect command interpretation; see Section 2.9.1.1 (on page 48).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

32842        **PWD**      This variable shall represent an absolute pathname of the current working directory. Assignments to this variable may be ignored unless the value is an absolute pathname of the current working directory and there are no filename components of dot or dot-dot.

## 32846 ASYNCHRONOUS EVENTS

32847        Default.

## 32848 STDOUT

32849        See the STDERR section.

## 32850 STDERR

32851        Except as otherwise stated (by the descriptions of any invoked utilities or in interactive mode), standard error shall be used only for diagnostic messages.

## 32853 OUTPUT FILES

32854        None.

## 32855 EXTENDED DESCRIPTION

32856        See Chapter 2. The following additional capabilities are supported on systems supporting the User Portability Utilities option.

### 32858 Command History List

32859        When the *sh* utility is being used interactively, it shall maintain a list of commands previously entered from the terminal in the file named by the *HISTFILE* environment variable. The type, size, and internal format of this file are unspecified. Multiple *sh* processes can share access to the file for a user, if file access permissions allow this; see the description of the *HISTFILE* environment variable.

### 32864 Command Line Editing

32865        When *sh* is being used interactively from a terminal, the current command and the command history (see *fc*) can be edited using vi-mode command line editing. This mode uses commands, described below, similar to a subset of those described in the *vi* utility. Implementations may offer other command line editing modes corresponding to other editing utilities.

32866        The command *set -o vi* shall enable vi-mode editing and place *sh* into *vi* insert mode (see **Command Line Editing (vi-mode)** (on page 852)). This command also shall disable any other editing mode that the implementation may provide. The command *set +o vi* disables vi-mode editing.

32867        Certain block-mode terminals may be unable to support shell command line editing. If a terminal is unable to provide either edit mode, it need not be possible to *set -o vi* when using the shell on this terminal.

32868        In the following sections, the characters *erase*, *interrupt*, *kill*, and *end-of-file* are those set by the *stty* utility.

32878       **Command Line Editing (vi-mode)**

32879       In *vi* editing mode, there shall be a distinguished line, the edit line. All the editing operations  
32880       which modify a line affect the edit line. The edit line is always the newest line in the command  
32881       history buffer.

32882       With *vi*-mode enabled, *sh* can be switched between insert mode and command mode.

32883       When in insert mode, an entered character shall be inserted into the command line, except as  
32884       noted in **vi Line Editing Insert Mode**. Upon entering *sh* and after termination of the previous  
32885       command, *sh* shall be in insert mode.

32886       Typing an escape character shall switch *sh* into command mode (see **vi Line Editing Command**  
32887       **Mode** (on page 853)). In command mode, an entered character shall either invoke a defined  
32888       operation, be used as part of a multi-character operation, or be treated as an error. A character  
32889       that is not recognized as part of an editing command shall terminate any specific editing  
32890       command and shall alert the terminal. Typing the *interrupt* character in command mode shall  
32891       cause *sh* to terminate command line editing on the current command line, reissue the prompt on  
32892       the next line of the terminal, and reset the command history (see *fc*) so that the most recently  
32893       executed command is the previous command (that is, the command that was being edited when  
32894       it was interrupted is not reentered into the history).

32895       In the following sections, the phrase “move the cursor to the beginning of the word” shall mean  
32896       “move the cursor to the first character of the current word” and the phrase “move the cursor to  
32897       the end of the word” shall mean “move the cursor to the last character of the current word”. The  
32898       phrase “beginning of the command line” indicates the point between the end of the prompt  
32899       string issued by the shell (or the beginning of the terminal line, if there is no prompt string) and  
32900       the first character of the command text.

32901       **vi Line Editing Insert Mode**

32902       While in insert mode, any character typed shall be inserted in the current command line, unless  
32903       it is from the following set.

32904       <newline>      Execute the current command line. If the current command line is not empty, this  
32905       line shall be entered into the command history (see *fc*).

32906       ~~erase~~      Delete the character previous to the current cursor position and move the current  
32907       cursor position back one character. In insert mode, characters shall be erased from  
32908       both the screen and the buffer when backspacing.

32909       ~~interrupt~~     Terminate command line editing with the same effects as described for  
32910       interrupting command mode; see **Command Line Editing (vi-mode)**.

32911       ~~kill~~        Clear all the characters from the input line.

32912       <control>-V   Insert the next character input, even if the character is otherwise a special insert  
32913       mode character.

32914       <control>-W   Delete the characters from the one preceding the cursor to the preceding word  
32915       boundary. The word boundary in this case is the closer to the cursor of either the  
32916       beginning of the line or a character that is in neither the **blank** nor **punct** character  
32917       classification of the current locale.

32918       ~~end-of-file~~   Interpreted as the end of input in *sh*. This interpretation shall occur only at the  
32919       beginning of an input line. If *end-of-file* is entered other than at the beginning of the  
32920       line, the results are unspecified.

32921 <ESC> Place *sh* into command mode.

### 32922 vi Line Editing Command Mode

32923 In command mode for the command line editing feature, decimal digits not beginning with 0  
32924 that precede a command letter shall be remembered. Some commands use these decimal digits  
32925 as a count number that affects the operation.

32926 The term *motion command* represents one of the commands:

32927 <space> 0 b F 1 w ^ \$ ; E f T w | , B e h t

32928 If the current line is not the edit line, any command that modifies the current line shall cause the  
32929 content of the current line to replace the content of the edit line, and the current line shall  
32930 become the edit line. This replacement cannot be undone (see the **u** and **U** commands below).  
32931 The modification requested shall then be performed to the edit line. When the current line is the  
32932 edit line, the modification shall be done directly to the edit line.

32933 Any command that is preceded by *count* shall take a count (the numeric value of any preceding  
32934 decimal digits). Unless otherwise noted, this count shall cause the specified operation to repeat  
32935 by the number of times specified by the count. Also unless otherwise noted, a *count* that is out of  
32936 range is considered an error condition and shall alert the terminal, but neither the cursor  
32937 position, nor the command line, shall change.

32938 The terms *word* and *bigword* are used as defined in the *vi* description. The term *save buffer*  
32939 corresponds to the term *unnamed buffer* in *vi*.

32940 The following commands shall be recognized in command mode:

32941 <newline> Execute the current command line. If the current command line is not empty, this  
32942 line shall be entered into the command history (see *fc*).

32943 <control>-L Redraw the current command line. Position the cursor at the same location on the  
32944 redrawn line.

32945 # Insert the character '#' at the beginning of the current command line and treat the  
32946 resulting edit line as a comment. This line shall be entered into the command  
32947 history; see *fc*.

32948 = Display the possible shell word expansions (see Section 2.6 (on page 36)) of the  
32949 bigword at the current command line position.

32950 **Note:** This does not modify the content of the current line, and therefore does not  
32951 cause the current line to become the edit line.

32952 These expansions shall be displayed on subsequent terminal lines. If the bigword  
32953 contains none of the characters '?', '\*', or '[', an asterisk ('\*') shall be  
32954 implicitly assumed at the end. If any directories are matched, these expansions  
32955 shall have a '/' character appended. After the expansion, the line shall be  
32956 redrawn, the cursor repositioned at the current cursor position, and *sh* shall be  
32957 placed in command mode.

32958 \ Perform pathname expansion (see Section 2.6.6 (on page 42)) on the current  
32959 bigword, up to the largest set of characters that can be matched uniquely. If the  
32960 bigword contains none of the characters '?', '\*', or '[', an asterisk ('\*') shall  
32961 be implicitly assumed at the end. This maximal expansion then shall replace the  
32962 original bigword in the command line, and the cursor shall be placed after this  
32963 expansion. If the resulting bigword completely and uniquely matches a directory, a  
32964 '/' character shall be inserted directly after the bigword. If some other file is  
32965 completely matched, a single <space> shall be inserted after the bigword. After

|       |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32966 |                         | this operation, <i>sh</i> shall be placed in insert mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 32967 | *                       | Perform pathname expansion on the current bigword and insert all expansions into the command to replace the current bigword, with each expansion separated by a single <space>. If at the end of the line, the current cursor position shall be moved to the first column position following the expansions and <i>sh</i> shall be placed in insert mode. Otherwise, the current cursor position shall be the last column position of the first character after the expansions and <i>sh</i> shall be placed in insert mode. If the current bigword contains none of the characters '?', '*', or '[', before the operation, an asterisk shall be implicitly assumed at the end.                                                           |
| 32975 | @ <i>letter</i>         | Insert the value of the alias named <i>_letter</i> . The symbol <i>letter</i> represents a single alphabetic character from the portable character set; implementations may support additional characters as an extension. If the alias <i>_letter</i> contains other editing commands, these commands shall be performed as part of the insertion. If no alias <i>_letter</i> is enabled, this command shall have no effect.                                                                                                                                                                                                                                                                                                             |
| 32980 | [ <i>count</i> ]~       | Convert, if the current character is a lowercase letter, to the equivalent uppercase letter and vice versa, as prescribed by the current locale. The current cursor position then shall be advanced by one character. If the cursor was positioned on the last character of the line, the case conversion shall occur, but the cursor shall not advance. If the '~' command is preceded by a <i>count</i> , that number of characters shall be converted, and the cursor shall be advanced to the character position after the last character converted. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line. |
| 32989 | [ <i>count</i> ].       | Repeat the most recent non-motion command, even if it was executed on an earlier command line. If the previous command was preceded by a <i>count</i> , and no count is given on the '.' command, the count from the previous command shall be included as part of the repeated command. If the '.' command is preceded by a <i>count</i> , this shall override any <i>count</i> argument to the previous command. The <i>count</i> specified in the '.' command shall become the count for subsequent '.' commands issued without a count.                                                                                                                                                                                               |
| 32996 | [ <i>number</i> ]v      | Invoke the vi editor to edit the current command line in a temporary file. When the editor exits, the commands in the temporary file shall be executed and placed in the command history. If a <i>number</i> is included, it specifies the command number in the command history to be edited, rather than the current command line.                                                                                                                                                                                                                                                                                                                                                                                                      |
| 33000 | [ <i>count</i> ]l (ell) | Move the current cursor position to the next character position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                                                                                                                                                                                                                                                                                     |
| 33001 | [ <i>count</i> ]<space> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33002 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33003 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33004 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33005 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33006 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33007 | [ <i>count</i> ]h       | Move the current cursor position to the <i>count</i> th (default 1) previous character position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the count is larger than the number of characters before the cursor, this shall not be considered an error; the cursor shall move to the first character on the line.                                                                                                                                                                                                                                                                                                                               |
| 33008 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33009 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33010 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33011 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33012 | [ <i>count</i> ]w       | Move to the start of the next word. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 33013 |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

|       |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33014 |           | advanced. If the <i>count</i> is larger than the number of words after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                                                                                                                                                                                               |
| 33015 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33016 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33017 | [count]W  | Move to the start of the next bigword. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of bigwords after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                           |
| 33018 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33019 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33020 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33021 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33022 | [count]e  | Move to the end of the current word. If at the end of a word, move to the end of the next word. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of words after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                     |
| 33023 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33024 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33025 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33026 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33027 | [count]E  | Move to the end of the current bigword. If at the end of a bigword, move to the end of the next bigword. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of bigwords after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                         |
| 33028 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33029 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33030 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33031 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33032 | [count]b  | Move to the beginning of the current word. If at the beginning of a word, move to the beginning of the previous word. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of words preceding the cursor, this shall not be considered an error; the cursor shall return to the first character on the line.             |
| 33033 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33034 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33035 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33036 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33037 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33038 | [count]B  | Move to the beginning of the current bigword. If at the beginning of a bigword, move to the beginning of the previous bigword. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of bigwords preceding the cursor, this shall not be considered an error; the cursor shall return to the first character on the line. |
| 33039 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33040 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33041 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33042 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33043 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33044 | ^         | Move the current cursor position to the first character on the input line that is not a <blank>.                                                                                                                                                                                                                                                                                                                                                    |
| 33045 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33046 | \$        | Move to the last character position on the current command line.                                                                                                                                                                                                                                                                                                                                                                                    |
| 33047 | 0         | (Zero.) Move to the first character position on the current command line.                                                                                                                                                                                                                                                                                                                                                                           |
| 33048 | [count]   | Move to the <i>count</i> th character position on the current command line. If no number is specified, move to the first position. The first character position shall be numbered 1. If the count is larger than the number of characters on the line, this shall not be considered an error; the cursor shall be placed on the last character on the line.                                                                                         |
| 33049 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33050 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33051 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33052 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33053 | [count]fc | Move to the first occurrence of the character 'c' that occurs after the current cursor position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the character 'c' does not occur in the line after the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                                            |
| 33054 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33055 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33056 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33057 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33058 | [count]Fc | Move to the first occurrence of the character 'c' that occurs before the current cursor position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the character 'c'                                                                                                                                                                                            |
| 33059 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33060 |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33061 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | does not occur in the line before the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                                                                                                                                                                                                                                                               |
| 33062 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33063 | [count]tc                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Move to the character before the first occurrence of the character 'c' that occurs after the current cursor position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the character 'c' does not occur in the line after the current cursor position, the terminal shall be alerted and the cursor shall not be moved. |
| 33064 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33065 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33066 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33067 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33068 | [count]Tc                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Move to the character after the first occurrence of the character 'c' that occurs before the current cursor position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the character 'c' does not occur in the line before the current cursor position, the terminal shall be alerted and the cursor shall not be moved.  |
| 33069 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33070 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33071 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33072 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33073 | [count];                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Repeat the most recent f, F, t, or T command. Any number argument on that previous command shall be ignored. Errors are those described for the repeated command.                                                                                                                                                                                                                                             |
| 33074 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33075 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33076 | [count],                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Repeat the most recent f, F, t, or T command. Any number argument on that previous command shall be ignored. However, reverse the direction of that command.                                                                                                                                                                                                                                                  |
| 33077 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33078 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33079 | a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Enter insert mode after the current cursor position. Characters that are entered shall be inserted before the next character.                                                                                                                                                                                                                                                                                 |
| 33080 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33081 | A                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Enter insert mode after the end of the current command line.                                                                                                                                                                                                                                                                                                                                                  |
| 33082 | i                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Enter insert mode at the current cursor position. Characters that are entered shall be inserted before the current character.                                                                                                                                                                                                                                                                                 |
| 33083 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33084 | I                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Enter insert mode at the beginning of the current command line.                                                                                                                                                                                                                                                                                                                                               |
| 33085 | R                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Enter insert mode, replacing characters from the command line beginning at the current cursor position.                                                                                                                                                                                                                                                                                                       |
| 33086 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33087 | [count]cmotion                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Delete the characters between the current cursor position and the cursor position that would result from the specified motion command. Then enter insert mode before the first character following any deleted characters. If count is specified, it shall be applied to the motion command. A count shall be ignored for the following motion commands:                                                      |
| 33088 | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33089 | ^                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33090 | \$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33091 | c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33092 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33093 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33094 | If the motion command is the character 'c', the current command line shall be cleared and insert mode shall be entered. If the motion command would move the current cursor position toward the beginning of the command line, the character under the current cursor position shall not be deleted. If the motion command would move the current cursor position toward the end of the command line, the character under the current cursor position shall be deleted. If the count is larger than the number of characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be deleted and insert mode shall be entered. If the motion command is invalid, the terminal shall be alerted, the cursor shall not be moved, and no text shall be deleted. |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33095 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33096 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33097 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33098 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33099 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33100 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33101 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33102 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33103 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33104 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33105 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                               |

|       |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33106 | <b>C</b>              | Delete from the current character to the end of the line and enter insert mode at the new end-of-line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33107 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33108 | <b>S</b>              | Clear the entire edit line and enter insert mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 33109 | <b>[count]rc</b>      | Replace the current character with the character ' <i>c</i> '. With a number <i>count</i> , replace the current and the following <i>count</i> -1 characters. After this command, the current cursor position shall be on the last character that was changed. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; all of the remaining characters shall be changed.                                                                                                                                                                                                                                                                                                                                                                      |
| 33110 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33111 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33112 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33113 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33114 | <b>[count]_</b>       | Append a <space> after the current character position and then append the last bigword in the previous input line after the <space>. Then enter insert mode after the last character just appended. With a number <i>count</i> , append the <i>count</i> th bigword in the previous line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 33115 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33116 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33117 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33118 | <b>[count]x</b>       | Delete the character at the current cursor position and place the deleted characters in the save buffer. If the cursor was positioned on the last character of the line, the character shall be deleted and the cursor position shall be moved to the previous character (the new last character). If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; all the characters from the cursor to the end of the line shall be deleted.                                                                                                                                                                                                                                                                                                        |
| 33119 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33120 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33121 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33122 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33123 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33124 | <b>[count]X</b>       | Delete the character before the current cursor position and place the deleted characters in the save buffer. The character under the current cursor position shall not change. If the cursor was positioned on the first character of the line, the terminal shall be alerted, and the X command shall have no effect. If the line contained a single character, the X command shall have no effect. If the line contained no characters, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of characters before the cursor, this shall not be considered an error; all the characters from before the cursor to the beginning of the line shall be deleted.                                                                                  |
| 33125 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33126 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33127 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33128 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33129 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33130 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33131 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33132 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33133 | <b>[count]dmotion</b> | Delete the characters between the current cursor position and the character position that would result from the motion command. A number <i>count</i> repeats the motion command <i>count</i> times. If the motion command would move toward the beginning of the command line, the character under the current cursor position shall not be deleted. If the motion command is d, the entire current command line shall be cleared. If the <i>count</i> is larger than the number of characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be deleted. The deleted characters shall be placed in the save buffer. |
| 33134 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33135 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33136 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33137 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33138 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33139 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33140 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33141 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33142 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33143 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33144 | <b>D</b>              | Delete all characters from the current cursor position to the end of the line. The deleted characters shall be placed in the save buffer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 33145 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33146 | <b>[count]ymotion</b> | Yank (that is, copy) the characters from the current cursor position to the position resulting from the motion command into the save buffer. A number <i>count</i> shall be applied to the motion command. If the motion command would move toward the beginning of the command line, the character under the current cursor position shall not be included in the set of yanked characters. If the motion command is y, the entire current command line shall be yanked into the save buffer. The current cursor position shall be unchanged. If the <i>count</i> is larger than the number of                                                                                                                                                                                                            |
| 33147 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33148 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33149 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33150 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33151 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33152 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33153 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|       |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33154 |                                | characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be yanked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 33155 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33156 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33157 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33158 | <b>Y</b>                       | Yank the characters from the current cursor position to the end of the line into the save buffer. The current character position shall be unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33159 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33160 | <b>[count]P</b>                | Put a copy of the current contents of the save buffer after the current cursor position. The current cursor position shall be advanced to the last character put from the save buffer. A <i>count</i> shall indicate how many copies of the save buffer shall be put.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 33161 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33162 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33163 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33164 | <b>[count]P</b>                | Put a copy of the current contents of the save buffer before the current cursor position. The current cursor position shall be moved to the last character put from the save buffer. A <i>count</i> shall indicate how many copies of the save buffer shall be put.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 33165 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33166 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33167 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33168 | <b>u</b>                       | Undo the last command that changed the edit line. This operation shall not undo the copy of any command line to the edit line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 33169 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33170 | <b>U</b>                       | Undo all changes made to the edit line. This operation shall not undo the copy of any command line to the edit line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 33171 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33172 | <b>[count]k</b>                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33173 | <b>[count]–</b>                | Set the current command line to be the <i>count</i> th previous command line in the shell command history. If <i>count</i> is not specified, it shall default to 1. The cursor shall be positioned on the first character of the new command. If a <b>k</b> or <b>–</b> command would retreat past the maximum number of commands in effect for this shell (affected by the <i>HISTSIZE</i> environment variable), the terminal shall be alerted, and the command shall have no effect.                                                                                                                                                                                                                                                                                                                        |
| 33174 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33175 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33176 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33177 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33178 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33179 | <b>[count]j</b>                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33180 | <b>[count]+</b>                | Set the current command line to be the <i>count</i> th next command line in the shell command history. If <i>count</i> is not specified, it shall default to 1. The cursor shall be positioned on the first character of the new command. If a <b>j</b> or <b>+</b> command advances past the edit line, the current command line shall be restored to the edit line and the terminal shall be alerted.                                                                                                                                                                                                                                                                                                                                                                                                        |
| 33181 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33182 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33183 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33184 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33185 | <b>[number]G</b>               | Set the current command line to be the oldest command line stored in the shell command history. With a number <i>number</i> , set the current command line to be the command line <i>number</i> in the history. If command line <i>number</i> does not exist, the terminal shall be alerted and the command line shall not be changed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 33186 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33187 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33188 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33189 | <b>/pattern&lt;newline&gt;</b> | Move backwards through the command history, searching for the specified pattern, beginning with the previous command line. Patterns use the pattern matching notation described in Section 2.13 (on page 62), except that the '^' character shall have special meaning when it appears as the first character of <i>pattern</i> . In this case, the '^' is discarded and the characters after the '^' shall be matched only at the beginning of a line. Commands in the command history shall be treated as strings, not as filenames. If the pattern is not found, the current command line shall be unchanged and the terminal is alerted. If it is found in a previous line, the current command line shall be set to that line and the cursor shall be set to the first character of the new command line. |
| 33190 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33191 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33192 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33193 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33194 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33195 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33196 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33197 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33198 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33199 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

33200 If *pattern* is empty, the last non-empty pattern provided to / or ? shall be used. If  
33201 there is no previous non-empty pattern, the terminal shall be alerted and the  
33202 current command line shall remain unchanged.

33203 **?pattern<newline>**  
33204 Move forwards through the command history, searching for the specified pattern,  
33205 beginning with the next command line. Patterns use the pattern matching notation  
33206 described in Section 2.13 (on page 62), except that the '^' character shall have  
33207 special meaning when it appears as the first character of *pattern*. In this case, the  
33208 '^' is discarded and the characters after the '^' shall be matched only at the  
33209 beginning of a line. Commands in the command history shall be treated as strings,  
33210 not as filenames. If the pattern is not found, the current command line shall be  
33211 unchanged and the terminal alerted. If it is found in a following line, the current  
33212 command line shall be set to that line and the cursor shall be set to the fist  
33213 character of the new command line.

33214 If *pattern* is empty, the last non-empty pattern provided to / or ? shall be used. If  
33215 there is no previous non-empty pattern, the terminal shall be alerted and the  
33216 current command line shall remain unchanged.

33217 **n** Repeat the most recent / or ? command. If there is no previous / or ?, the terminal  
33218 shall be alerted and the current command line shall remain unchanged.  
33219 **N** Repeat the most recent / or ? command, reversing the direction of the search. If  
33220 there is no previous / or ?, the terminal shall be alerted and the current command  
33221 line shall remain unchanged.

## 33222 EXIT STATUS

33223 The following exit values shall be returned:

33224 0 The script to be executed consisted solely of zero or more blank lines or comments, or  
33225 both.  
33226 1-125 A non-interactive shell detected a syntax, redirection, or variable assignment error.  
33227 127 A specified *command\_file* could not be found by a non-interactive shell.

33228 Otherwise, the shell shall return the exit status of the last command it invoked or attempted to  
33229 invoke (see also the *exit* utility in Section 2.14 (on page 64)).

## 33230 CONSEQUENCES OF ERRORS

33231 See Section 2.8.1 (on page 46).

## 33232 APPLICATION USAGE

33233 Standard input and standard error are the files that determine whether a shell is interactive  
33234 when -i is not specified. For example:

33235 sh > file

33236 and:

33237 sh 2> file

33238 create interactive and non-interactive shells, respectively. Although both accept terminal input,  
33239 the results of error conditions are different, as described in Section 2.8.1 (on page 46); in the  
33240 second example a redirection error encountered by a special built-in utility aborts the shell.

33241 A conforming application must protect its first operand, if it starts with a plus sign, by preceding  
33242 it with the "--" argument that denotes the end of the options.

33243 Applications should note that the standard *PATH* to the shell cannot be assumed to be either  
33244 **/bin/sh** or **/usr/bin/sh**, and should be determined by interrogation of the *PATH* returned by  
33245 *getconf PATH*, ensuring that the returned pathname is an absolute pathname and not a shell  
33246 built-in.

33247 For example, to determine the location of the standard *sh* utility:

33248 command -v sh

33249 On some implementations this might return:

33250 /usr/xpg4/bin/sh

33251 Furthermore, on systems that support executable scripts (the "#!" construct), it is  
33252 recommended that applications using executable scripts install them using *getconf -v* to  
33253 determine the shell pathname and update the "#!" script appropriately as it is being installed  
33254 (for example, with *sed*). For example:

```

Installation time script to install correct POSIX shell pathname

Get list of paths to check

Sifs=$IFS
IFS=:
set $(getconf PATH)
IFS=$Sifs

Check each path for 'sh'

for i in $@
do
 if [-f ${i}/sh];
 then
 Pshell=${i}/sh
 fi
done

This is the list of scripts to update. They should be of the
form '${name}.source' and will be transformed to '${name}'.
Each script should begin:

!INSTALLSHELLPATH -p

scripts="a b c"

Transform each script

for i in ${scripts}
do
 sed -e "s|${INSTALLSHELLPATH}|${Pshell}|" < ${i}.source > ${i}
done
```

## 33289 EXAMPLES

- 33290        1. Execute a shell command from a string:  
33291              sh -c "cat myfile"
- 33292        2. Execute a shell script from a file in the current directory:  
33293              sh my\_shell\_cmds

## 33294 RATIONALE

33295        The *sh* utility and the *set* special built-in utility share a common set of options.

33296        The KornShell ignores the contents of *IFS* upon entry to the script. A conforming application  
33297        cannot rely on importing *IFS*. One justification for this, beyond security considerations, is to  
33298        assist possible future shell compilers. Allowing *IFS* to be imported from the environment  
33299        prevents many optimizations that might otherwise be performed via dataflow analysis of the  
33300        script itself.

33301        The text in the STDIN section about non-blocking reads concerns an instance of *sh* that has been  
33302        invoked, probably by a C-language program, with standard input that has been opened using  
33303        the O\_NONBLOCK flag; see *open()* in the System Interfaces volume of IEEE Std 1003.1-2001. If  
33304        the shell did not reset this flag, it would immediately terminate because no input data would be  
33305        available yet and that would be considered the same as end-of-file.

33306        The options associated with a *restricted shell* (command name *rsh* and the *-r* option) were  
33307        excluded because the standard developers considered that the implied level of security could  
33308        not be achieved and they did not want to raise false expectations.

33309        On systems that support set-user-ID scripts, a historical trapdoor has been to link a script to the  
33310        name *-i*. When it is called by a sequence such as:

33311        sh -  
33312        or by:  
33313        #! usr/bin/sh -

33314        the historical systems have assumed that no option letters follow. Thus, this volume of  
33315        IEEE Std 1003.1-2001 allows the single hyphen to mark the end of the options, in addition to the  
33316        use of the regular "--" argument, because it was considered that the older practice was so  
33317        pervasive. An alternative approach is taken by the KornShell, where real and effective  
33318        user/group IDs must match for an interactive shell; this behavior is specifically allowed by this  
33319        volume of IEEE Std 1003.1-2001.

33320        **Note:** There are other problems with set-user-ID scripts that the two approaches described here do  
33321        not resolve.

33322        The initialization process for the history file can be dependent on the system start-up files, in  
33323        that they may contain commands that effectively preempt the user's settings of *HISTFILE* and  
33324        *HISTSIZE*. For example, function definition commands are recorded in the history file, unless  
33325        the *set -o nolog* option is set. If the system administrator includes function definitions in some  
33326        system start-up file called before the *ENV* file, the history file is initialized before the user gets a  
33327        chance to influence its characteristics. In some historical shells, the history file is initialized just  
33328        after the *ENV* file has been processed. Therefore, it is implementation-defined whether changes  
33329        made to *HISTFILE* after the history file has been initialized are effective.

33330        The default messages for the various *MAIL*-related messages are unspecified because they vary  
33331        across implementations. Typical messages are:

33332 "you have mail\n"

33333 or:

33334 "you have new mail\n"

33335 It is important that the descriptions of command line editing refer to the same shell as that in  
33336 IEEE Std 1003.1-2001 so that interactive users can also be application programmers without  
33337 having to deal with programmatic differences in their two environments. It is also essential that  
33338 the utility name *sh* be specified because this explicit utility name is too firmly rooted in historical  
33339 practice of application programs for it to change.

33340 Consideration was given to mandating a diagnostic message when attempting to set *vi*-mode on  
33341 terminals that do not support command line editing. However, it is not historical practice for the  
33342 shell to be cognizant of all terminal types and thus be able to detect inappropriate terminals in  
33343 all cases. Implementations are encouraged to supply diagnostics in this case whenever possible,  
33344 rather than leaving the user in a state where editing commands work incorrectly.

33345 In early proposals, the KornShell-derived *emacs* mode of command line editing was included,  
33346 even though the *emacs* editor itself was not. The community of *emacs* proponents was adamant  
33347 that the full *emacs* editor not be standardized because they were concerned that an attempt to  
33348 standardize this very powerful environment would encourage vendors to ship strictly  
33349 conforming versions lacking the extensibility required by the community. The author of the  
33350 original *emacs* program also expressed his desire to omit the program. Furthermore, there were a  
33351 number of historical systems that did not include *emacs*, or included it without supporting it, but  
33352 there were very few that did not include and support *vi*. The shell *emacs* command line editing  
33353 mode was finally omitted because it became apparent that the KornShell version and the editor  
33354 being distributed with the GNU system had diverged in some respects. The author of *emacs*  
33355 requested that the POSIX *emacs* mode either be deleted or have a significant number of  
33356 unspecified conditions. Although the KornShell author agreed to consider changes to bring the  
33357 shell into alignment, the standard developers decided to defer specification at that time. At the  
33358 time, it was assumed that convergence on an acceptable definition would occur for a subsequent  
33359 draft, but that has not happened, and there appears to be no impetus to do so. In any case,  
33360 implementations are free to offer additional command line editing modes based on the exact  
33361 models of editors their users are most comfortable with.

33362 Early proposals had the following list entry in **vi Line Editing Insert Mode** (on page 852):

33363 \ If followed by the *erase* or *kill* character, that character shall be inserted into the input line.  
33364 Otherwise, the backslash itself shall be inserted into the input line.

33365 However, this is not actually a feature of *sh* command line editing insert mode, but one of some  
33366 historical terminal line drivers. Some conforming implementations continue to do this when the  
33367 **stty iexten** flag is set.

## 33368 FUTURE DIRECTIONS

33369 None.

## 33370 SEE ALSO

33371 Chapter 2 (on page 29), *cd*, *echo*, *exit*, *fc*, *pwd*, *read*, *set*, *stty*, *test*, *umask*, *vi*, the System Interfaces  
33372 volume of IEEE Std 1003.1-2001, *dup()*, *exec*, *exit()*, *fork()*, *open()*, *pipe()*, *signal()*, *system()*,  
33373 *ulimit()*, *umask()*, *wait()*

## 33374 CHANGE HISTORY

33375 First released in Issue 2.

**33376 Issue 5**

- 33377 The FUTURE DIRECTIONS section is added.
- 33378 Text is added to the DESCRIPTION for the Large File Summit proposal.

**33379 Issue 6**

- 33380 The Open Group Corrigendum U029/2 is applied, correcting the second SYNOPSIS.
- 33381 The Open Group Corrigendum U027/3 is applied, correcting a typographical error.
- 33382 The following new requirements on POSIX implementations derive from alignment with the Single UNIX Specification:
- The option letters derived from the *set* special built-in are also accepted with a leading plus sign ('+').
  - Large file extensions are added:
    - Pathname expansion does not fail due to the size of a file.
    - Shell input and output redirections have an implementation-defined offset maximum that is established in the open file description.
- 33389 In the ENVIRONMENT VARIABLES section, the text “user’s home directory” is updated to “directory referred to by the *HOME* environment variable”.
- 33392 Descriptions for the *ENV* and *PWD* environment variables are included to align with the IEEE P1003.2b draft standard.
- 33394 The normative text is reworded to avoid use of the term “must” for application requirements.

**33395 NAME**

33396        sleep — suspend execution for an interval

**33397 SYNOPSIS**

33398        sleep *time*

**33399 DESCRIPTION**

33400        The *sleep* utility shall suspend execution for at least the integral number of seconds specified by  
33401        the *time* operand.

**33402 OPTIONS**

33403        None.

**33404 OPERANDS**

33405        The following operand shall be supported:

33406        *time*        A non-negative decimal integer specifying the number of seconds for which to  
33407        suspend execution.

**33408 STDIN**

33409        Not used.

**33410 INPUT FILES**

33411        None.

**33412 ENVIRONMENT VARIABLES**

33413        The following environment variables shall affect the execution of *sleep*:

33414        *LANG*      Provide a default value for the internationalization variables that are unset or null.  
33415                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
33416                  Internationalization Variables for the precedence of internationalization variables  
33417                  used to determine the values of locale categories.)

33418        *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
33419                  internationalization variables.

33420        *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
33421                  characters (for example, single-byte as opposed to multi-byte characters in  
33422                  arguments).

**33423        *LC\_MESSAGES***

33424                  Determine the locale that should be used to affect the format and contents of  
33425                  diagnostic messages written to standard error.

33426 XSI        *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**33427 ASYNCHRONOUS EVENTS**

33428        If the *sleep* utility receives a SIGALRM signal, one of the following actions shall be taken:

- 33429        1. Terminate normally with a zero exit status.
- 33430        2. Effectively ignore the signal.
- 33431        3. Provide the default behavior for signals described in the ASYNCHRONOUS EVENTS  
33432                  section of Section 1.11 (on page 20). This could include terminating with a non-zero exit  
33433                  status.

33434        The *sleep* utility shall take the standard action for all other signals.

## 33435 STDOUT

33436 Not used.

33437 STDERR

33438 The standard error shall be used only for diagnostic messages.

33439 OUTPUT FILES

33440 None.

## **33441 EXTENDED DESCRIPTION**

33442                  None.

## 33443 EXIT STATUS

33444 The following exit values shall be returned:

33445 0 The execution was successfully suspended for at least *time* seconds, or a SIGALRM signal  
33446 was received. See the ASYNCHRONOUS EVENTS section.

33447 >0 An error occurred.

## **33448 CONSEQUENCES OF ERRORS**

33449 Default.

## **33450 APPLICATION USAGE**

33451                  None.

33452 EXAMPLES

33453 The *sleep* utility can be used to execute a command after a certain amount of time, as in:

33454 (sleep 105; command) &

33455 or to execute a command every so often, as in:

```
33456 while true
33457 do
33458 command
33459 sleep 1
33460 done
```

33461 RATIONALE

The exit status is allowed to be zero when *sleep* is interrupted by the SIGALRM signal because most implementations of this utility rely on the arrival of that signal to notify them that the requested finishing time has been successfully attained. Such implementations thus do not distinguish this situation from the successful completion case. Other implementations are allowed to catch the signal and go back to sleep until the requested time expires or to provide the normal signal termination procedures.

As with all other utilities that take integral operands and do not specify subranges of allowed values, *sleep* is required by this volume of IEEE Std 1003.1-2001 to deal with *time* requests of up to 2 147 483 647 seconds. This may mean that some implementations have to make multiple calls to the delay mechanism of the underlying operating system if its argument range is less than this.

## 33473 FUTURE DIRECTIONS

33474                  None.

33475 SEE ALSO

33476      *wait*, the System Interfaces volume of IEEE Std 1003.1-2001, *alarm()*, *sleep()*

**33477 CHANGE HISTORY**

33478 First released in Issue 2.

## 33479 NAME

33480 sort — sort, merge, or sequence check text files

## 33481 SYNOPSIS

33482 sort [-m] [-o *output*] [-bdfinru] [-t *char*] [-k *keydef*]... [*file...*]

33483 sort -c [-bdfinru] [-t *char*] [-k *keydef*] [*file*]

## 33484 DESCRIPTION

33485 The *sort* utility shall perform one of the following functions:

- 33486 1. Sort lines of all the named files together and write the result to the specified output.
- 33487 2. Merge lines of all the named (presorted) files together and write the result to the specified output.
- 33488 3. Check that a single input file is correctly presorted.

33490 Comparisons shall be based on one or more sort keys extracted from each line of input (or, if no  
33491 sort keys are specified, the entire line up to, but not including, the terminating <newline>), and  
33492 shall be performed using the collating sequence of the current locale.

## 33493 OPTIONS

33494 The *sort* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
33495 12.2, Utility Syntax Guidelines, and the **-k** *keydef* option should follow the **-b**, **-d**, **-f**, **-i**, **-n**, and  
33496 **-r** options.

33497 The following options shall be supported:

- 33498 **-c** Check that the single input file is ordered as specified by the arguments and the  
33499 collating sequence of the current locale. No output shall be produced; only the exit  
33500 code shall be affected.
- 33501 **-m** Merge only; the input file shall be assumed to be already sorted.
- 33502 **-o *output*** Specify the name of an output file to be used instead of the standard output. This  
33503 file can be the same as one of the input files.
- 33504 **-u** Unique: suppress all but one in each set of lines having equal keys. If used with  
33505 the **-c** option, check that there are no lines with duplicate keys, in addition to  
33506 checking that the input file is sorted.

33507 The following options shall override the default ordering rules. When ordering options appear  
33508 independent of any key field specifications, the requested field ordering rules shall be applied  
33509 globally to all sort keys. When attached to a specific key (see **-k**), the specified ordering options  
33510 shall override all global ordering options for that key.

- 33511 **-d** Specify that only <blank>s and alphanumeric characters, according to the current  
33512 setting of *LC\_CTYPE*, shall be significant in comparisons. The behavior is  
33513 undefined for a sort key to which **-i** or **-n** also applies.
- 33514 **-f** Consider all lowercase characters that have uppercase equivalents, according to  
33515 the current setting of *LC\_CTYPE*, to be the uppercase equivalent for the purposes  
33516 of comparison.
- 33517 **-i** Ignore all characters that are non-printable, according to the current setting of  
33518 *LC\_CTYPE*.
- 33519 **-n** Restrict the sort key to an initial numeric string, consisting of optional <blank>s,  
33520 optional minus sign, and zero or more digits with an optional radix character and  
33521 thousands separators (as defined in the current locale), which shall be sorted by

33522 arithmetic value. An empty digit string shall be treated as zero. Leading zeros and  
 33523 signs on zeros shall not affect ordering.

33524 **-r** Reverse the sense of comparisons.

33525 The treatment of field separators can be altered using the options:

33526 **-b** Ignore leading <blank>s when determining the starting and ending positions of a  
 33527 restricted sort key. If the **-b** option is specified before the first **-k** option, it shall be  
 33528 applied to all **-k** options. Otherwise, the **-b** option can be attached independently  
 33529 to each **-k field\_start** or **field\_end** option-argument (see below).

33530 **-t char** Use *char* as the field separator character; *char* shall not be considered to be part of a  
 33531 field (although it can be included in a sort key). Each occurrence of *char* shall be  
 33532 significant (for example, <*char*><*char*> delimits an empty field). If **-t** is not  
 33533 specified, <blank>s shall be used as default field separators; each maximal non-  
 33534 empty sequence of <blank>s that follows a non-<blank> shall be a field separator.

33535 Sort keys can be specified using the options:

33536 **-k keydef** The *keydef* argument is a restricted sort key field definition. The format of this  
 33537 definition is:

33538 *field\_start[type][,field\_end[type]]*

33539 where *field\_start* and *field\_end* define a key field restricted to a portion of the line  
 33540 (see the EXTENDED DESCRIPTION section), and *type* is a modifier from the list of  
 33541 characters 'b', 'd', 'f', 'i', 'n', 'r'. The 'b' modifier shall behave like the  
 33542 **-b** option, but shall apply only to the *field\_start* or *field\_end* to which it is attached.  
 33543 The other modifiers shall behave like the corresponding options, but shall apply  
 33544 only to the key field to which they are attached; they shall have this effect if  
 33545 specified with *field\_start*, *field\_end*, or both. If any modifier is attached to a  
 33546 *field\_start* or to a *field\_end*, no option shall apply to either. Implementations shall  
 33547 support at least nine occurrences of the **-k** option, which shall be significant in  
 33548 command line order. If no **-k** option is specified, a default sort key of the entire  
 33549 line shall be used.

33550 When there are multiple key fields, later keys shall be compared only after all  
 33551 earlier keys compare equal. Except when the **-u** option is specified, lines that  
 33552 otherwise compare equal shall be ordered as if none of the options **-d**, **-f**, **-i**, **-n**, or  
 33553 **-k** were present (but with **-r** still in effect, if it was specified) and with all bytes in  
 33554 the lines significant to the comparison. The order in which lines that still compare  
 33555 equal are written is unspecified.

## 33556 OPERANDS

33557 The following operand shall be supported:

33558 **file** A pathname of a file to be sorted, merged, or checked. If no *file* operands are  
 33559 specified, or if a *file* operand is '**-**', the standard input shall be used.

## 33560 STDIN

33561 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '**-**'.  
 33562 See the INPUT FILES section.

## 33563 INPUT FILES

33564 The input files shall be text files, except that the *sort* utility shall add a <newline> to the end of a  
 33565 file ending with an incomplete last line.

**33566 ENVIRONMENT VARIABLES**

33567 The following environment variables shall affect the execution of *sort*:

- 33568 **LANG** Provide a default value for the internationalization variables that are unset or null.  
33569 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
33570 Internationalization Variables for the precedence of internationalization variables  
33571 used to determine the values of locale categories.)
- 33572 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
33573 internationalization variables.
- 33574 **LC\_COLLATE**  
33575 Determine the locale for ordering rules.
- 33576 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
33577 characters (for example, single-byte as opposed to multi-byte characters in  
33578 arguments and input files) and the behavior of character classification for the **-b**,  
33579 **-d**, **-f**, **-i**, and **-n** options.
- 33580 **LC\_MESSAGES**  
33581 Determine the locale that should be used to affect the format and contents of  
33582 diagnostic messages written to standard error.
- 33583 **LC\_NUMERIC**  
33584 Determine the locale for the definition of the radix character and thousands  
33585 separator for the **-n** option.
- 33586 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**33587 ASYNCHRONOUS EVENTS**

33588 Default.

**33589 STDOUT**

33590 Unless the **-o** or **-c** options are in effect, the standard output shall contain the sorted input.

**33591 STDERR**

33592 The standard error shall be used for diagnostic messages. A warning message about correcting  
33593 an incomplete last line of an input file may be generated, but need not affect the final exit status.

**33594 OUTPUT FILES**

33595 If the **-o** option is in effect, the sorted input shall be written to the file *output*.

**33596 EXTENDED DESCRIPTION**

33597 The notation:

33598 **-k** *field\_start[type]*[**,** *field\_end[type]*]  
33599 shall define a key field that begins at *field\_start* and ends at *field\_end* inclusive, unless *field\_start*  
33600 falls beyond the end of the line or after *field\_end*, in which case the key field is empty. A missing  
33601 *field\_end* shall mean the last character of the line.

33602 A field comprises a maximal sequence of non-separating characters and, in the absence of option  
33603 **-t**, any preceding field separator.

33604 The *field\_start* portion of the *keydef* option-argument shall have the form:

33605 *field\_number*[**.** *first\_character*]

33606 Fields and characters within fields shall be numbered starting with 1. The *field\_number* and  
33607 *first\_character* pieces, interpreted as positive decimal integers, shall specify the first character to  
33608 be used as part of a sort key. If *first\_character* is omitted, it shall refer to the first character of the

33609 field.

33610 The *field\_end* portion of the *keydef* option-argument shall have the form:

33611 *field\_number*[.*last\_character*]

33612 The *field\_number* shall be as described above for *field\_start*. The *last\_character* piece, interpreted  
33613 as a non-negative decimal integer, shall specify the last character to be used as part of the sort  
33614 key. If *last\_character* evaluates to zero or *.last\_character* is omitted, it shall refer to the last  
33615 character of the field specified by *field\_number*.

33616 If the **-b** option or **b** type modifier is in effect, characters within a field shall be counted from the  
33617 first non-<blank> in the field. (This shall apply separately to *first\_character* and *last\_character*.)

### 33618 EXIT STATUS

33619 The following exit values shall be returned:

- 33620 0 All input files were output successfully, or **-c** was specified and the input file was correctly  
33621 sorted.
- 33622 1 Under the **-c** option, the file was not ordered as specified, or if the **-c** and **-u** options were  
33623 both specified, two input lines were found with equal keys.
- 33624 >1 An error occurred.

### 33625 CONSEQUENCES OF ERRORS

33626 Default.

### 33627 APPLICATION USAGE

33628 The default value for **-t**, <blank>, has different properties from, for example, **-t "<space>"**. If a  
33629 line contains:

33630 <space><space>foo

33631 the following treatment would occur with default separation as opposed to specifically selecting  
33632 a <space>:

| Field | Default           | <b>-t "&lt;space&gt;"</b> |
|-------|-------------------|---------------------------|
| 1     | <space><space>foo | <i>empty</i>              |
| 2     | <i>empty</i>      | <i>empty</i>              |
| 3     | <i>empty</i>      | foo                       |

33637 The leading field separator itself is included in a field when **-t** is not used. For example, this  
33638 command returns an exit status of zero, meaning the input was already sorted:

```
33639 sort -c -k 2 <<eof
33640 y<tab>b
33641 x<space>a
33642 eof
```

33643 (assuming that a <tab> precedes the <space> in the current collating sequence). The field  
33644 separator is not included in a field when it is explicitly set via **-t**. This is historical practice and  
33645 allows usage such as:

```
33646 sort -t " | " -k 2n <<eof
33647 Atlanta|425022|Georgia
33648 Birmingham|284413|Alabama
33649 Columbia|100385|South Carolina
33650 eof
```

33651 where the second field can be correctly sorted numerically without regard to the non-numeric  
33652 field separator.

33653 The wording in the OPTIONS section clarifies that the **-b**, **-d**, **-f**, **-i**, **-n**, and **-r** options have to  
33654 come before the first sort key specified if they are intended to apply to all specified keys. The  
33655 way it is described in this volume of IEEE Std 1003.1-2001 matches historical practice, not  
33656 historical documentation. The results are unspecified if these options are specified after a **-k**  
33657 option.

33658 The **-f** option might not work as expected in locales where there is not a one-to-one mapping  
33659 between an uppercase and a lowercase letter.

## 33660 EXAMPLES

1. The following command sorts the contents of **infile** with the second field as the sort key:

33662       sort -k 2,2 infile

2. The following command sorts, in reverse order, the contents of **infile1** and **infile2**, placing  
33664 the output in **outfile** and using the second character of the second field as the sort key  
33665 (assuming that the first character of the second field is the field separator):

33666       sort -r -o outfile -k 2.2,2.2 infile1 infile2

3. The following command sorts the contents of **infile1** and **infile2** using the second non-  
33668 <blank> of the second field as the sort key:

33669       sort -k 2.2b,2.2b infile1 infile2

4. The following command prints the System V password file (user database) sorted by the  
33671 numeric user ID (the third colon-separated field):

33672       sort -t : -k 3,3n /etc/passwd

5. The following command prints the lines of the already sorted file **infile**, suppressing all  
33674 but one occurrence of lines having the same third field:

33675       sort -um -k 3.1,3.0 infile

## 33676 RATIONALE

33677 Examples in some historical documentation state that options **-um** with one input file keep the  
33678 first in each set of lines with equal keys. This behavior was deemed to be an implementation  
33679 artifact and was not standardized.

33680 The **-z** option was omitted; it is not standard practice on most systems and is inconsistent with  
33681 using **sort** to sort several files individually and then merge them together. The text concerning **-z**  
33682 in historical documentation appeared to require implementations to determine the proper buffer  
33683 length during the sort phase of operation, but not during the merge.

33684 The **-y** option was omitted because of non-portability. The **-M** option, present in System V, was  
33685 omitted because of non-portability in international usage.

33686 An undocumented **-T** option exists in some implementations. It is used to specify a directory for  
33687 intermediate files. Implementations are encouraged to support the use of the **TMPDIR**  
33688 environment variable instead of adding an option to support this functionality.

33689 The **-k** option was added to satisfy two objections. First, the zero-based counting used by **sort** is  
33690 not consistent with other utility conventions. Second, it did not meet syntax guideline  
33691 requirements.

33692 Historical documentation indicates that “setting **-n** implies **-b**”. The description of **-n** already  
33693 states that optional leading <blank>s are tolerated in doing the comparison. If **-b** is enabled,

33694 rather than implied, by **-n**, this has unusual side effects. When a character offset is used in a  
33695 column of numbers (for example, to sort modulo 100), that offset is measured relative to the  
33696 most significant digit, not to the column. Based upon a recommendation from the author of the  
33697 original *sort* utility, the **-b** implication has been omitted from this volume of  
33698 IEEE Std 1003.1-2001, and an application wishing to achieve the previously mentioned side  
33699 effects has to code the **-b** flag explicitly.

### 33700 **FUTURE DIRECTIONS**

33701 None.

### 33702 **SEE ALSO**

33703 *comm*, *join*, *uniq*, the System Interfaces volume of IEEE Std 1003.1-2001, *toupper()*

### 33704 **CHANGE HISTORY**

33705 First released in Issue 2.

### 33706 **Issue 6**

33707 IEEE PASC Interpretation 1003.2 #174 is applied, updating the DESCRIPTION of comparisons.

33708 IEEE PASC Interpretation 1003.2 #168 is applied.

## 33709 NAME

33710 split — split files into pieces

## 33711 SYNOPSIS

33712 UP `split [-l line_count][-a suffix_length][file[name]]`

33713 `split -b n[k|m][-a suffix_length][file[name]]`

33714

## 33715 DESCRIPTION

33716 The *split* utility shall read an input file and write one or more output files. The default size of  
 33717 each output file shall be 1 000 lines. The size of the output files can be modified by specification  
 33718 of the **-b** or **-l** options. Each output file shall be created with a unique suffix. The suffix shall  
 33719 consist of exactly *suffix\_length* lowercase letters from the POSIX locale. The letters of the suffix  
 33720 shall be used as if they were a base-26 digit system, with the first suffix to be created consisting  
 33721 of all 'a' characters, the second with a 'b' replacing the last 'a', and so on, until a name of all  
 33722 'z' characters is created. By default, the names of the output files shall be 'x', followed by a  
 33723 two-character suffix from the character set as described above, starting with "aa", "ab", "ac",  
 33724 and so on, and continuing until the suffix "zz", for a maximum of 676 files.

33725 If the number of files required exceeds the maximum allowed by the suffix length provided,  
 33726 such that the last allowable file would be larger than the requested size, the *split* utility shall fail  
 33727 after creating the last file with a valid suffix; *split* shall not delete the files it created with valid  
 33728 suffixes. If the file limit is not exceeded, the last file created shall contain the remainder of the  
 33729 input file, and may be smaller than the requested size.

## 33730 OPTIONS

33731 The *split* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 33732 12.2, Utility Syntax Guidelines.

33733 The following options shall be supported:

33734 **-a** *suffix\_length*

33735 Use *suffix\_length* letters to form the suffix portion of the filenames of the split file.  
 33736 If **-a** is not specified, the default suffix length shall be two. If the sum of the *name*  
 33737 operand and the *suffix\_length* option-argument would create a filename exceeding  
 33738 *{NAME\_MAX}* bytes, an error shall result; *split* shall exit with a diagnostic  
 33739 message and no files shall be created.

33740 **-b** *n* Split a file into pieces *n* bytes in size.

33741 **-b** *nk* Split a file into pieces *n*\*1 024 bytes in size.

33742 **-b** *nm* Split a file into pieces *n*\*1 048 576 bytes in size.

33743 **-l** *line\_count* Specify the number of lines in each resulting file piece. The *line\_count* argument is  
 33744 an unsigned decimal integer. The default is 1 000. If the input does not end with a  
 33745 <newline>, the partial line shall be included in the last output file.

## 33746 OPERANDS

33747 The following operands shall be supported:

33748 *file* The pathname of the ordinary file to be split. If no input file is given or *file* is '**-**',  
 33749 the standard input shall be used.

33750 *name* The prefix to be used for each of the files resulting from the split operation. If no  
 33751 *name* argument is given, 'x' shall be used as the prefix of the output files. The  
 33752 combined length of the basename of *prefix* and *suffix\_length* cannot exceed  
 33753 *{NAME\_MAX}* bytes. See the OPTIONS section.

**33754 STDIN**

33755 See the INPUT FILES section.

**33756 INPUT FILES**

33757 Any file can be used as input.

**33758 ENVIRONMENT VARIABLES**

33759 The following environment variables shall affect the execution of *split*:

33760 **LANG** Provide a default value for the internationalization variables that are unset or null.  
33761 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
33762 Internationalization Variables for the precedence of internationalization variables  
33763 used to determine the values of locale categories.)

33764 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
33765 internationalization variables.

33766 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
33767 characters (for example, single-byte as opposed to multi-byte characters in  
33768 arguments and input files).

**33769 *LC\_MESSAGES***

33770 Determine the locale that should be used to affect the format and contents of  
33771 diagnostic messages written to standard error.

33772 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**33773 ASYNCHRONOUS EVENTS**

33774 Default.

**33775 STDOUT**

33776 Not used.

**33777 STDERR**

33778 The standard error shall be used only for diagnostic messages.

**33779 OUTPUT FILES**

33780 The output files contain portions of the original input file; otherwise, unchanged.

**33781 EXTENDED DESCRIPTION**

33782 None.

**33783 EXIT STATUS**

33784 The following exit values shall be returned:

33785 0 Successful completion.

33786 >0 An error occurred.

**33787 CONSEQUENCES OF ERRORS**

33788 Default.

**33789 APPLICATION USAGE**

33790 None.

**33791 EXAMPLES**

33792 In the following examples **foo** is a text file that contains 5 000 lines.

- 33793 1. Create five files, **xaa**, **xab**, **xac**, **xad**, and **xae**:

33794 `split foo`

- 33795 2. Create five files, but the suffixed portion of the created files consists of three letters, **xaaa**,  
33796 **xaab**, **xaac**, **xaad**, and **xaae**:

33797 `split -a 3 foo`

- 33798 3. Create three files with four-letter suffixes and a supplied prefix, **bar\_aaaa**, **bar\_aaab**, and  
33799 **bar\_aaac**:

33800 `split -a 4 -l 2000 foo bar_`

- 33801 4. Create as many files as are necessary to contain at most 20\*1 024 bytes, each with the  
33802 default prefix of **x** and a five-letter suffix:

33803 `split -a 5 -b 20k foo`

**33804 RATIONALE**

33805 The **-b** option was added to provide a mechanism for splitting files other than by lines. While  
33806 most uses of the **-b** option are for transmitting files over networks, some believed it would have  
33807 additional uses.

33808 The **-a** option was added to overcome the limitation of being able to create only 676 files.

33809 Consideration was given to deleting this utility, using the rationale that the functionality  
33810 provided by this utility is available via the *csplit* utility (see *csplit*). Upon reconsideration of the  
33811 purpose of the User Portability Extension, it was decided to retain both this utility and the *csplit*  
33812 utility because users use both utilities and have historical expectations of their behavior.  
33813 Furthermore, the splitting on byte boundaries in *split* cannot be duplicated with the historical  
33814 *csplit*.

33815 The text “*split* shall not delete the files it created with valid suffixes” would normally be  
33816 assumed, but since the related utility, *csplit*, does delete files under some circumstances, the  
33817 historical behavior of *split* is made explicit to avoid misinterpretation.

**33818 FUTURE DIRECTIONS**

33819 None.

**33820 SEE ALSO**

33821 *csplit*

**33822 CHANGE HISTORY**

33823 First released in Issue 2.

**33824 Issue 6**

33825 This utility is marked as part of the User Portability Utilities option.

33826 The APPLICATION USAGE section is added.

33827 The obsolescent SYNOPSIS is removed.

## 33828 NAME

33829 strings — find printable strings in files

## 33830 SYNOPSIS

33831 UP strings [-a][-t *format*][-n *number*][*file...*]

33832

## 33833 DESCRIPTION

33834 The *strings* utility shall look for printable strings in regular files and shall write those strings to  
33835 standard output. A printable string is any sequence of four (by default) or more printable  
33836 characters terminated by a <newline> or NUL character. Additional implementation-defined  
33837 strings may be written; see *localedef*.

## 33838 OPTIONS

33839 The *strings* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
33840 12.2, Utility Syntax Guidelines.

33841 The following options shall be supported:

33842 **-a** Scan files in their entirety. If **-a** is not specified, it is implementation-defined what  
33843 portion of each file is scanned for strings.

33844 **-n** *number* Specify the minimum string length, where the *number* argument is a positive  
33845 decimal integer. The default shall be 4.

33846 **-t** *format* Write each string preceded by its byte offset from the start of the file. The format  
33847 shall be dependent on the single character used as the *format* option-argument:

33848     d The offset shall be written in decimal.

33849     o The offset shall be written in octal.

33850     x The offset shall be written in hexadecimal.

## 33851 OPERANDS

33852 The following operand shall be supported:

33853 **file** A pathname of a regular file to be used as input. If no *file* operand is specified, the  
33854 *strings* utility shall read from the standard input.

## 33855 STDIN

33856 See the INPUT FILES section.

## 33857 INPUT FILES

33858 The input files named by the utility arguments or the standard input shall be regular files of any  
33859 format.

## 33860 ENVIRONMENT VARIABLES

33861 The following environment variables shall affect the execution of *strings*:

33862 **LANG** Provide a default value for the internationalization variables that are unset or null.  
33863 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
33864 Internationalization Variables for the precedence of internationalization variables  
33865 used to determine the values of locale categories.)

33866 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
33867 internationalization variables.

33868 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
33869 characters (for example, single-byte as opposed to multi-byte characters in  
33870 arguments and input files) and to identify printable strings.

- 33871       ***LC\_MESSAGES***  
33872              Determine the locale that should be used to affect the format and contents of  
33873              diagnostic messages written to standard error.
- 33874 XSI       **NLSPATH**     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 33875 **ASYNCHRONOUS EVENTS**
- 33876              Default.
- 33877 **STDOUT**
- 33878              Strings found shall be written to the standard output, one per line.
- 33879              When the **-t** option is not specified, the format of the output shall be:  
33880              "%s", <string>
- 33881              With the **-t o** option, the format of the output shall be:  
33882              "%o %s", <byte offset>, <string>
- 33883              With the **-t x** option, the format of the output shall be:  
33884              "%x %s", <byte offset>, <string>
- 33885              With the **-t d** option, the format of the output shall be:  
33886              "%d %s", <byte offset>, <string>
- 33887 **STDERR**
- 33888              The standard error shall be used only for diagnostic messages.
- 33889 **OUTPUT FILES**
- 33890              None.
- 33891 **EXTENDED DESCRIPTION**
- 33892              None.
- 33893 **EXIT STATUS**
- 33894              The following exit values shall be returned:
- 33895              0   Successful completion.
- 33896              >0   An error occurred.
- 33897 **CONSEQUENCES OF ERRORS**
- 33898              Default.
- 33899 **APPLICATION USAGE**
- 33900              By default the data area (as opposed to the text, "bss", or header areas) of a binary executable  
33901              file is scanned. Implementations document which areas are scanned.
- 33902              Some historical implementations do not require NUL or <newline> terminators for strings to  
33903              permit those languages that do not use NUL as a string terminator to have their strings written.
- 33904 **EXAMPLES**
- 33905              None.
- 33906 **RATIONALE**
- 33907              Apart from rationalizing the option syntax and slight difficulties with object and executable  
33908              binary files, *strings* is specified to match historical practice closely. The **-a** and **-n** options were  
33909              introduced to replace the non-conforming **-** and **-number** options.
- 33910              The **-o** option historically means different things on different implementations. Some use it to  
33911              mean "*offset* in decimal", while others use it as "*offset* in octal". Instead of trying to decide which

33912 way would be least objectionable, the **-t** option was added. It was originally named **-O** to mean  
33913 “offset”, but was changed to **-t** to be consistent with *od*.

33914 The ISO C standard function *isprint()* is restricted to a domain of **unsigned char**. This volume of  
33915 IEEE Std 1003.1-2001 requires implementations to write strings as defined by the current locale.

### 33916 FUTURE DIRECTIONS

33917 None.

### 33918 SEE ALSO

33919 *localedef, nm*

### 33920 CHANGE HISTORY

33921 First released in Issue 4.

### 33922 Issue 6

33923 This utility is marked as part of the User Portability Utilities option.

33924 The obsolescent SYNOPSIS is removed.

33925 The normative text is reworded to avoid use of the term “must” for application requirements.

**33926 NAME**

33927        strip — remove unnecessary information from executable files (**DEVELOPMENT**)

**33928 SYNOPSIS**

33929 SD        `strip file...`

33930

**33931 DESCRIPTION**

33932        The *strip* utility shall remove from executable files named by the *file* operands any information  
33933        the implementor deems unnecessary for execution of those files. The nature of that information  
33934        is unspecified. The effect of *strip* shall be similar to the use of the *-s* option to *c99* or *fort77*.

**33935 OPTIONS**

33936        None.

**33937 OPERANDS**

33938        The following operand shall be supported:

33939        *file*        A pathname referring to an executable file.

**33940 STDIN**

33941        Not used.

**33942 INPUT FILES**

33943        The input files shall be in the form of executable files successfully produced by any compiler  
33944        defined by this volume of IEEE Std 1003.1-2001.

**33945 ENVIRONMENT VARIABLES**

33946        The following environment variables shall affect the execution of *strip*:

33947        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
33948                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
33949                  Internationalization Variables for the precedence of internationalization variables  
33950                  used to determine the values of locale categories.)

33951        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
33952                  internationalization variables.

33953        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
33954                  characters (for example, single-byte as opposed to multi-byte characters in  
33955                  arguments).

33956        *LC\_MESSAGES*

33957                  Determine the locale that should be used to affect the format and contents of  
33958                  diagnostic messages written to standard error.

33959 XSI        *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**33960 ASYNCHRONOUS EVENTS**

33961        Default.

**33962 STDOUT**

33963        Not used.

**33964 STDERR**

33965        The standard error shall be used only for diagnostic messages.

**33966 OUTPUT FILES**

33967       The *strip* utility shall produce executable files of unspecified format.

**33968 EXTENDED DESCRIPTION**

33969       None.

**33970 EXIT STATUS**

33971       The following exit values shall be returned:

33972       0   Successful completion.

33973       >0   An error occurred.

**33974 CONSEQUENCES OF ERRORS**

33975       Default.

**33976 APPLICATION USAGE**

33977       None.

**33978 EXAMPLES**

33979       None.

**33980 RATIONALE**

33981       Historically, this utility has been used to remove the symbol table from an executable file. It was  
33982       included since it is known that the amount of symbolic information can amount to several  
33983       megabytes; the ability to remove it in a portable manner was deemed important, especially for  
33984       smaller systems.

33985       The behavior of *strip* is said to be the same as the *-s* option to a compiler. While the end result is  
33986       essentially the same, it is not required to be identical.

**33987 FUTURE DIRECTIONS**

33988       None.

**33989 SEE ALSO**

33990       *ar*, *c99*, *fort77*

**33991 CHANGE HISTORY**

33992       First released in Issue 2.

**33993 Issue 6**

33994       This utility is marked as part of the Software Development Utilities option.

## 33995 NAME

33996        stty — set the options for a terminal

## 33997 SYNOPSIS

33998        stty [ -a | -g ]

33999        stty *operands*

## 34000 DESCRIPTION

34001        The *stty* utility shall set or report on terminal I/O characteristics for the device that is its  
34002 standard input. Without options or operands specified, it shall report the settings of certain  
34003 characteristics, usually those that differ from implementation-defined defaults. Otherwise, it  
34004 shall modify the terminal state according to the specified operands. Detailed information about  
34005 the modes listed in the first five groups below are described in the Base Definitions volume of  
34006 IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. Operands in the Combination  
34007 Modes group (see **Combination Modes** (on page 886)) are implemented using operands in the  
34008 previous groups. Some combinations of operands are mutually-exclusive on some terminal  
34009 types; the results of using such combinations are unspecified.

34010        Typical implementations of this utility require a communications line configured to use the  
34011 **termios** interface defined in the System Interfaces volume of IEEE Std 1003.1-2001. On systems  
34012 where none of these lines are available, and on lines not currently configured to support the  
34013 **termios** interface, some of the operands need not affect terminal characteristics.

## 34014 OPTIONS

34015        The *stty* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
34016 12.2, Utility Syntax Guidelines.

34017        The following options shall be supported:

34018        **-a**        Write to standard output all the current settings for the terminal.

34019        **-g**        Write to standard output all the current settings in an unspecified form that can be  
34020 used as arguments to another invocation of the *stty* utility on the same system. The  
34021 form used shall not contain any characters that would require quoting to avoid  
34022 word expansion by the shell; see Section 2.6 (on page 36).

## 34023 OPERANDS

34024        The following operands shall be supported to set the terminal characteristics.

## 34025        Control Modes

34026        **parenb** (**-parenb**) Enable (disable) parity generation and detection. This shall have the effect of  
34027 setting (not setting) PARENB in the **termios c\_cflag** field, as defined in the  
34028 Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General  
34029 Terminal Interface.

34030        **parodd** (**-parodd**)

34031        Select odd (even) parity. This shall have the effect of setting (not setting)  
34032 PARODD in the **termios c\_cflag** field, as defined in the Base Definitions  
34033 volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.

34034        **cs5 cs6 cs7 cs8**

34035        Select character size, if possible. This shall have the effect of setting CS5, CS6,  
34036 CS7, and CS8, respectively, in the **termios c\_cflag** field, as defined in the Base  
34037 Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal  
Interface.

34038        *number*

34039        Set terminal baud rate to the number given, if possible. If the baud rate is set  
to zero, the modem control lines shall no longer be asserted. This shall have

|       |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
|-------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34040 |                                  | the effect of setting the input and output <b>termios</b> baud rate values as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                                                                                         |
| 34041 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34042 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34043 | <b>ispeed</b> <i>number</i>      | Set terminal input baud rate to the number given, if possible. If the input baud rate is set to zero, the input baud rate shall be specified by the value of the output baud rate. This shall have the effect of setting the input <b>termios</b> baud rate values as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. |
| 34044 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34045 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34046 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34047 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34048 | <b>ospeed</b> <i>number</i>      | Set terminal output baud rate to the number given, if possible. If the output baud rate is set to zero, the modem control lines shall no longer be asserted. This shall have the effect of setting the output <b>termios</b> baud rate values as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                      |
| 34049 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34050 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34051 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34052 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34053 | <b>hupcl</b> ( <b>-hupcl</b> )   | Stop asserting modem control lines (do not stop asserting modem control lines) on last close. This shall have the effect of setting (not setting) HUPCL in the <b>termios</b> <i>c_cflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                |
| 34054 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34055 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34056 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34057 | <b>hup</b> ( <b>-hup</b> )       | Equivalent to <b>hupcl</b> ( <b>-hupcl</b> ).                                                                                                                                                                                                                                                                                                                                 |
| 34058 | <b>cstopb</b> ( <b>-cstopb</b> ) | Use two (one) stop bits per character. This shall have the effect of setting (not setting) CSTOPB in the <b>termios</b> <i>c_cflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                      |
| 34059 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34060 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34061 | <b>cread</b> ( <b>-cread</b> )   | Enable (disable) the receiver. This shall have the effect of setting (not setting) CREAD in the <b>termios</b> <i>c_cflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                               |
| 34062 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34063 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34064 | <b>clocal</b> ( <b>-clocal</b> ) | Assume a line without (with) modem control. This shall have the effect of setting (not setting) CLOCAL in the <b>termios</b> <i>c_cflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                 |
| 34065 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34066 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34067 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34068 |                                  | It is unspecified whether <i>stty</i> shall report an error if an attempt to set a Control Mode fails.                                                                                                                                                                                                                                                                        |
| 34069 | <b>Input Modes</b>               |                                                                                                                                                                                                                                                                                                                                                                               |
| 34070 | <b>ignbrk</b> ( <b>-ignbrk</b> ) | Ignore (do not ignore) break on input. This shall have the effect of setting (not setting) IGNBRK in the <b>termios</b> <i>c_iflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                      |
| 34071 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34072 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34073 | <b>brkint</b> ( <b>-brkint</b> ) | Signal (do not signal) INTR on break. This shall have the effect of setting (not setting) BRKINT in the <b>termios</b> <i>c_iflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                       |
| 34074 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34075 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34076 | <b>ignpar</b> ( <b>-ignpar</b> ) | Ignore (do not ignore) bytes with parity errors. This shall have the effect of setting (not setting) IGNPAR in the <b>termios</b> <i>c_iflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                            |
| 34077 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34078 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34079 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34080 | <b>parmrk</b> ( <b>-parmrk</b> ) | Mark (do not mark) parity errors. This shall have the effect of setting (not setting) PARMRK in the <b>termios</b> <i>c_iflag</i> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                           |
| 34081 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34082 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34083 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |
| 34084 |                                  |                                                                                                                                                                                                                                                                                                                                                                               |

|           |                         |                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34085     | <b>inpck (-inpck)</b>   | Enable (disable) input parity checking. This shall have the effect of setting (not setting) INPCK in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                |
| 34086     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34087     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34088     | <b>istrip (-strip)</b>  | Strip (do not strip) input characters to seven bits. This shall have the effect of setting (not setting) ISTRIP in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                  |
| 34089     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34090     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34091     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34092     | <b>inlcr (-inlcr)</b>   | Map (do not map) NL to CR on input. This shall have the effect of setting (not setting) INLCR in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                    |
| 34093     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34094     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34095     | <b>igncr (-igncr)</b>   | Ignore (do not ignore) CR on input. This shall have the effect of setting (not setting) IGNCR in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                    |
| 34096     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34097     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34098     | <b>icrnl (-icrnl)</b>   | Map (do not map) CR to NL on input. This shall have the effect of setting (not setting) ICRNL in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                    |
| 34099     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34100     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34101     | <b>ixon (-ixon)</b>     | Enable (disable) START/STOP output control. Output from the system is stopped when the system receives STOP and started when the system receives START. This shall have the effect of setting (not setting) IXON in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. |
| 34102     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34103     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34104     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34105     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34106 XSI | <b>ixany (-ixany)</b>   | Allow any character to restart output. This shall have the effect of setting (not setting) IXANY in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                 |
| 34107     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34108     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34109     | <b>ixoff (-ixoff)</b>   | Request that the system send (not send) STOP characters when the input queue is nearly full and START characters to resume data transmission. This shall have the effect of setting (not setting) IXOFF in the <b>termios c_iflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.          |
| 34110     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34111     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34112     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34113     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34114     | <b>Output Modes</b>     |                                                                                                                                                                                                                                                                                                                                                                  |
| 34115     | <b>opost (-opost)</b>   | Post-process output (do not post-process output; ignore all other output modes). This shall have the effect of setting (not setting) OPOST in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                       |
| 34116     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34117     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34118     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34119 XSI | <b>ocrnl (-ocrnl)</b>   | Map (do not map) CR to NL on output. This shall have the effect of setting (not setting) OCRNL in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                   |
| 34120     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34121     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34122     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34123     | <b>onocr (-onocr)</b>   | Do not (do) output CR at column zero. This shall have the effect of setting (not setting) ONOCR in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                  |
| 34124     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34125     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34126     | <b>onlret (-onlret)</b> | The terminal newline key performs (does not perform) the CR function. This shall have the effect of setting (not setting) ONLRET in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                 |
| 34127     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34128     |                         |                                                                                                                                                                                                                                                                                                                                                                  |
| 34129     |                         |                                                                                                                                                                                                                                                                                                                                                                  |

|       |                                  |                                                                                                                                                                                                                                                                                                                                                        |
|-------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34130 | <b>ofill</b> ( <b>-ofill</b> )   | Use fill characters (use timing) for delays. This shall have the effect of setting (not setting) OFILL in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                 |
| 34134 | <b>ofdel</b> ( <b>-ofdel</b> )   | Fill characters are DELs (NULs). This shall have the effect of setting (not setting) OFDEL in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                             |
| 34137 | <b>cr0 cr1 cr2 cr3</b>           | Select the style of delay for CRs. This shall have the effect of setting CRDLY to CR0, CR1, CR2, or CR3, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                 |
| 34141 | <b>nl0 nl1</b>                   | Select the style of delay for NL. This shall have the effect of setting NLDLY to NL0 or NL1, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                             |
| 34145 | <b>tab0 tab1 tab2 tab3</b>       | Select the style of delay for horizontal tabs. This shall have the effect of setting TABDLY to TAB0, TAB1, TAB2, or TAB3, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. Note that TAB3 has the effect of expanding <tab>s to <space>s. |
| 34151 | <b>tabs</b> ( <b>-tabs</b> )     | Synonym for <b>tab0 (tab3)</b> .                                                                                                                                                                                                                                                                                                                       |
| 34152 | <b>bs0 bs1</b>                   | Select the style of delay for backspaces. This shall have the effect of setting BSDLY to BS0 or BS1, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                     |
| 34156 | <b>ff0 ff1</b>                   | Select the style of delay for form-feeds. This shall have the effect of setting FFDLY to FF0 or FF1, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                     |
| 34160 | <b>vt0 vt1</b>                   | Select the style of delay for vertical-tabs. This shall have the effect of setting VTDLY to VT0 or VT1, respectively, in the <b>termios c_oflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                  |
| 34164 | <b>Local Modes</b>               |                                                                                                                                                                                                                                                                                                                                                        |
| 34165 | <b>isig</b> ( <b>-isig</b> )     | Enable (disable) the checking of characters against the special control characters INTR, QUIT, and SUSP. This shall have the effect of setting (not setting) ISIG in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                      |
| 34169 | <b>icanon</b> ( <b>-icanon</b> ) | Enable (disable) canonical input (ERASE and KILL processing). This shall have the effect of setting (not setting) ICANON in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                               |
| 34173 | <b>iexten</b> ( <b>-iexten</b> ) | Enable (disable) any implementation-defined special control characters not currently controlled by <b>icanon</b> , <b>isig</b> , <b>ixon</b> , or <b>ixoff</b> . This shall have the effect of setting (not setting) IEXTEN in the <b>termios c_lflag</b> field, as defined in the Base                                                                |

|       |                         |                                                                                                                                                                                                                                                                                                                                      |
|-------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34176 |                         | Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                                                                                                                                                                                  |
| 34177 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34178 | <b>echo (-echo)</b>     | Echo back (do not echo back) every character typed. This shall have the effect of setting (not setting) ECHO in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                         |
| 34179 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34180 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34181 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34182 | <b>echoe (-echoe)</b>   | The ERASE character visually erases (does not erase) the last character in the current line from the display, if possible. This shall have the effect of setting (not setting) ECHOE in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface. |
| 34183 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34184 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34185 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34186 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34187 | <b>echok (-echok)</b>   | Echo (do not echo) NL after KILL character. This shall have the effect of setting (not setting) ECHOK in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                |
| 34188 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34189 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34190 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34191 | <b>echonl (-echonl)</b> | Echo (do not echo) NL, even if <b>echo</b> is disabled. This shall have the effect of setting (not setting) ECHONL in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                   |
| 34192 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34193 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34194 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34195 | <b>noflsh (-noflsh)</b> | Disable (enable) flush after INTR, QUIT, SUSP. This shall have the effect of setting (not setting) NOFLSH in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                            |
| 34196 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34197 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34198 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34199 | <b>tostop (-tostop)</b> | Send SIGTTOU for background output. This shall have the effect of setting (not setting) TOSTOP in the <b>termios c_lflag</b> field, as defined in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface.                                                                                       |
| 34200 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34201 |                         |                                                                                                                                                                                                                                                                                                                                      |
| 34202 |                         |                                                                                                                                                                                                                                                                                                                                      |

### 34203 Special Control Character Assignments

34204 <*control>-character string*

34205 Set <*control>-character* to *string*. If <*control>-character* is one of the character sequences in  
34206 the first column of the following table, the corresponding Base Definitions volume of  
34207 IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface control character from the  
34208 second column shall be recognized. This has the effect of setting the corresponding element  
34209 of the **termios c\_cc** array (see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter  
34210 13, Headers, <termios.h>).

34211

**Table 4-19** Control Character Names in *stty*

| Control Character | c_cc Subscript | Description     |
|-------------------|----------------|-----------------|
| <b>eof</b>        | VEOF           | EOF character   |
| <b>eol</b>        | VEOL           | EOL character   |
| <b>erase</b>      | VERASE         | ERASE character |
| <b>intr</b>       | VINTR          | INTR character  |
| <b>kill</b>       | VKILL          | KILL character  |
| <b>quit</b>       | VQUIT          | QUIT character  |
| <b>susp</b>       | VSUSP          | SUSP character  |
| <b>start</b>      | VSTART         | START character |
| <b>stop</b>       | VSTOP          | STOP character  |

If *string* is a single character, the control character shall be set to that character. If *string* is the two-character sequence " ^- " or the string *undef*, the control character shall be set to \_POSIX\_VDISABLE , if it is in effect for the device; if \_POSIX\_VDISABLE is not in effect for the device, it shall be treated as an error. In the POSIX locale, if *string* is a two-character sequence beginning with circumflex (' ^ '), and the second character is one of those listed in the " ^c " column of the following table, the control character shall be set to the corresponding character value in the Value column of the table.

34229

**Table 4-20** Circumflex Control Characters in *stty*

| ^c   | Value | ^c   | Value | ^c   | Value |
|------|-------|------|-------|------|-------|
| a, A | <SOH> | l, L | <FF>  | w, W | <ETB> |
| b, B | <STX> | m, M | <CR>  | x, X | <CAN> |
| c, C | <ETX> | n, N | <SO>  | y, Y | <EM>  |
| d, D | <EOT> | o, O | <SI>  | z, Z | <SUB> |
| e, E | <ENQ> | p, P | <DLE> | [    | <ESC> |
| f, F | <ACK> | q, Q | <DC1> | \    | <FS>  |
| g, G | <BEL> | r, R | <DC2> | ]    | <GS>  |
| h, H | <BS>  | s, S | <DC3> | ^    | <RS>  |
| i, I | <HT>  | t, T | <DC4> | —    | <US>  |
| j, J | <LF>  | u, U | <NAK> | ?    | <DEL> |
| k, K | <VT>  | v, V | <SYN> |      |       |

**min number**

Set the value of MIN to *number*. MIN is used in non-canonical mode input processing (*icanon*).

**time number**

Set the value of TIME to *number*. TIME is used in non-canonical mode input processing (*icanon*).

**Combination Modes****saved settings**

Set the current terminal characteristics to the saved settings produced by the **-g** option.

**evenp or parity**

Enable **parenb** and **cs7**; disable **parodd**.

**oddp**

Enable **parenb**, **cs7**, and **parodd**.

|           |                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34255     | <b>-parity, -evenp, or -oddp</b>                                                                                                                                                                                                                                                                                                                                                     |
| 34256     | Disable <b>parenb</b> , and set <b>cs8</b> .                                                                                                                                                                                                                                                                                                                                         |
| 34257 XSI | <b>raw (-raw or cooked)</b>                                                                                                                                                                                                                                                                                                                                                          |
| 34258     | Enable (disable) raw input and output. Raw mode shall be equivalent to setting:                                                                                                                                                                                                                                                                                                      |
| 34259     | <code>stty cs8 erase ^- kill ^- intr ^- \<br/>quit ^- eof ^- eol ^- -post -inpck</code>                                                                                                                                                                                                                                                                                              |
| 34260     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34261     | <b>nl (-nl)</b>                                                                                                                                                                                                                                                                                                                                                                      |
| 34262     | Enable (disable) <b>icnrl</b> . In addition, <b>-nl</b> unsets <b>inlcr</b> and <b>igncr</b> .                                                                                                                                                                                                                                                                                       |
| 34263     | <b>ek</b> Reset ERASE and KILL characters back to system defaults.                                                                                                                                                                                                                                                                                                                   |
| 34264     | <b>sane</b>                                                                                                                                                                                                                                                                                                                                                                          |
| 34265     | Reset all modes to some reasonable, unspecified, values.                                                                                                                                                                                                                                                                                                                             |
| 34266     | <b>STDIN</b>                                                                                                                                                                                                                                                                                                                                                                         |
| 34267     | Although no input is read from standard input, standard input shall be used to get the current terminal I/O characteristics and to set new terminal I/O characteristics.                                                                                                                                                                                                             |
| 34268     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34269     | <b>INPUT FILES</b>                                                                                                                                                                                                                                                                                                                                                                   |
| 34270     | None.                                                                                                                                                                                                                                                                                                                                                                                |
| 34271     | <b>ENVIRONMENT VARIABLES</b>                                                                                                                                                                                                                                                                                                                                                         |
| 34272     | The following environment variables shall affect the execution of <b>stty</b> :                                                                                                                                                                                                                                                                                                      |
| 34273     | <b>LANG</b> Provide a default value for the internationalization variables that are unset or null.<br>(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)                                                                 |
| 34274     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34275     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34276     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34277     | <b>LC_ALL</b> If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                               |
| 34278     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34279     | <b>LC_CTYPE</b> This variable determines the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments) and which characters are in the class <b>print</b> .                                                                                                                         |
| 34280     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34281     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34282     | <b>LC_MESSAGES</b>                                                                                                                                                                                                                                                                                                                                                                   |
| 34283     | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                         |
| 34284     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34285 XSI | <b>NLSPATH</b> Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                                                                                                                                                                                                                                                                 |
| 34286     | <b>ASYNCHRONOUS EVENTS</b>                                                                                                                                                                                                                                                                                                                                                           |
| 34287     | Default.                                                                                                                                                                                                                                                                                                                                                                             |
| 34288     | <b>STDOUT</b>                                                                                                                                                                                                                                                                                                                                                                        |
| 34289     | If operands are specified, no output shall be produced.                                                                                                                                                                                                                                                                                                                              |
| 34290     | If the <b>-g</b> option is specified, <b>stty</b> shall write to standard output the current settings in a form that can be used as arguments to another instance of <b>stty</b> on the same system.                                                                                                                                                                                 |
| 34291     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34292     | If the <b>-a</b> option is specified, all of the information as described in the OPERANDS section shall be written to standard output. Unless otherwise specified, this information shall be written as <b>&lt;space&gt;</b> -separated tokens in an unspecified format, on one or more lines, with an unspecified number of tokens per line. Additional information may be written. |
| 34293     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34294     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34295     |                                                                                                                                                                                                                                                                                                                                                                                      |
| 34296     | If no options or operands are specified, an unspecified subset of the information written for the <b>-a</b> option shall be written.                                                                                                                                                                                                                                                 |
| 34297     |                                                                                                                                                                                                                                                                                                                                                                                      |

34298 If speed information is written as part of the default output, or if the **-a** option is specified and if  
34299 the terminal input speed and output speed are the same, the speed information shall be written  
34300 as follows:

34301 "speed %d baud; ", <speed>

34302 Otherwise, speeds shall be written as:

34303 "ispeed %d baud; ospeed %d baud; ", <ispeed>, <ospeed>

34304 In locales other than the POSIX locale, the word **baud** may be changed to something more  
34305 appropriate in those locales.

34306 If control characters are written as part of the default output, or if the **-a** option is specified,  
34307 control characters shall be written as:

34308 "%s = %s; ", <control-character name>, <value>

34309 where <value> is either the character, or some visual representation of the character if it is non-  
34310 printable, or the string *undef* if the character is disabled.

### 34311 STDRERR

34312 The standard error shall be used only for diagnostic messages.

### 34313 OUTPUT FILES

34314 None.

### 34315 EXTENDED DESCRIPTION

34316 None.

### 34317 EXIT STATUS

34318 The following exit values shall be returned:

34319 0 The terminal options were read or set successfully.

34320 >0 An error occurred.

### 34321 CONSEQUENCES OF ERRORS

34322 Default.

### 34323 APPLICATION USAGE

34324 The **-g** flag is designed to facilitate the saving and restoring of terminal state from the shell level.

34325 For example, a program may:

```
34326 saveterm="$(stty -g)" # save terminal state
34327 stty (new settings) # set new state
34328 ... # ...
34329 stty $saveterm # restore terminal state
```

34330 Since the format is unspecified, the saved value is not portable across systems.

34331 Since the **-a** format is so loosely specified, scripts that save and restore terminal settings should  
34332 use the **-g** option.

### 34333 EXAMPLES

34334 None.

### 34335 RATIONALE

34336 The original *stty* description was taken directly from System V and reflected the System V  
34337 terminal driver **termio**. It has been modified to correspond to the terminal driver **termios**.

34338 Output modes are specified only for XSI-conformant systems. All implementations are expected  
34339 to provide *stty* operands corresponding to all of the output modes they support.

34340 The **stty** utility is primarily used to tailor the user interface of the terminal, such as selecting the  
34341 preferred ERASE and KILL characters. As an application programming utility, **stty** can be used  
34342 within shell scripts to alter the terminal settings for the duration of the script.

34343 The **termios** section states that individual disabling of control characters is possible through the  
34344 option `_POSIX_VDISABLE`. If enabled, two conventions currently exist for specifying this:  
34345 System V uses "`^-`", and BSD uses `undef`. Both are accepted by **stty** in this volume of  
34346 IEEE Std 1003.1-2001. The other BSD convention of using the letter '`u`' was rejected because it  
34347 conflicts with the actual letter '`u`', which is an acceptable value for a control character.

34348 Early proposals did not specify the mapping of "`^c`" to control characters because the control  
34349 characters were not specified in the POSIX locale character set description file requirements. The  
34350 control character set is now specified in the Base Definitions volume of IEEE Std 1003.1-2001,  
34351 Chapter 3, Definitions so the historical mapping is specified. Note that although the mapping  
34352 corresponds to control-character key assignments on many terminals that use the  
34353 ISO/IEC 646: 1991 standard (or ASCII) character encodings, the mapping specified here is to the  
34354 control characters, not their keyboard encodings.

34355 Since **termios** supports separate speeds for input and output, two new options were added to  
34356 specify each distinctly.

34357 Some historical implementations use standard input to get and set terminal characteristics;  
34358 others use standard output. Since input from a login TTY is usually restricted to the owner while  
34359 output to a TTY is frequently open to anyone, using standard input provides fewer chances of  
34360 accidentally (or maliciously) altering the terminal settings of other users. Using standard input  
34361 also allows **stty -a** and **stty -g** output to be redirected for later use. Therefore, usage of standard  
34362 input is required by this volume of IEEE Std 1003.1-2001.

### 34363 FUTURE DIRECTIONS

34364 None.

### 34365 SEE ALSO

34366 Chapter 2 (on page 29), the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11,  
34367 General Terminal Interface, <**termios.h**>

### 34368 CHANGE HISTORY

34369 First released in Issue 2.

#### 34370 Issue 5

34371 The description of **tabs** is clarified.

34372 The FUTURE DIRECTIONS section is added.

#### 34373 Issue 6

34374 The legacy items **iuclc(–iuclc)**, **xcase**, **olcuc(–olcuc)**, **lcase(–lcase)**, and **LCASE(–LCASE)** are  
34375 removed.

## 34376 NAME

34377 tabs — set terminal tabs

## 34378 SYNOPSIS

34379 UP XSI tabs [-n | -a | -a2 | -c | -c2 | -c3 | -f | -p | -s | -u][+m[n]] [-T type]

34380 tabs [-T type][+[n]] n1[,n2,...]

34381

## 34382 DESCRIPTION

34383 The *tabs* utility shall display a series of characters that first clears the hardware terminal tab settings and then initializes the tab stops at the specified positions and optionally adjusts the margin.

34386 The phrase “tab-stop position *N*” shall be taken to mean that, from the start of a line of output, tabbing to position *N* shall cause the next character output to be in the (*N*+1)th column position on that line. The maximum number of tab stops allowed is terminal-dependent.

34389 It need not be possible to implement *tabs* on certain terminals. If the terminal type obtained from the *TERM* environment variable or *-T* option represents such a terminal, an appropriate diagnostic message shall be written to standard error and *tabs* shall exit with a status greater than zero.

## 34393 OPTIONS

34394 The *tabs* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines, except for various extensions: the options **-a2**, **-c2**, and **-c3** are multi-character.

34397 The following options shall be supported:

34398 **-n** Specify repetitive tab stops separated by a uniform number of column positions, *n*, where *n* is a single-digit decimal number. The default usage of *tabs* with no arguments shall be equivalent to **tabs-8**. When **-0** is used, the tab stops shall be cleared and no new ones set.

34402 XSI **-a** 1,10,16,36,72  
34403 Assembler, applicable to some mainframes.

34404 XSI **-a2** 1,10,16,40,72  
34405 Assembler, applicable to some mainframes.

34406 XSI **-c** 1,8,12,16,20,55  
34407 COBOL, normal format.

34408 XSI **-c2** 1,6,10,14,49  
34409 COBOL, compact format (columns 1 to 6 omitted).

34410 XSI **-c3** 1,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,67  
34411 COBOL compact format (columns 1 to 6 omitted), with more tabs than **-c2**.

34412 XSI **-f** 1,7,11,15,19,23  
34413 FORTRAN

34414 XSI **-p** 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61  
34415 PL/1

34416 XSI **-s** 1,10,55  
34417 SNOBOL

34418 XSI **-u** 1,12,20,44  
34419 Assembler, applicable to some mainframes.

34420        **-T type**      Indicate the type of terminal. If this option is not supplied and the *TERM* variable is unset or null, an unspecified default terminal type shall be used. The setting of *type* shall take precedence over the value in *TERM*.

### 34423 OPERANDS

34424        The following operand shall be supported:

34425        **n1[,n2,...]**     A single command line argument that consists of tab-stop values separated using either commas or <blank>s. The application shall ensure that the tab-stop values are positive decimal integers in strictly ascending order. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. For example, the tab lists 1,10,20,30 and 1,10,+10,+10 are considered to be identical.

### 34431 STDIN

34432        Not used.

### 34433 INPUT FILES

34434        None.

### 34435 ENVIRONMENT VARIABLES

34436        The following environment variables shall affect the execution of *tabs*:

34437        **LANG**        Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

34441        **LC\_ALL**      If set to a non-empty string value, override the values of all the other internationalization variables.

34443        **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

#### 34446 **LC\_MESSAGES**

34447        Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

34449 XSI     **NLSPATH**    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

34450        **TERM**        Determine the terminal type. If this variable is unset or null, and if the **-T** option is not specified, an unspecified default terminal type shall be used.

### 34452 ASYNCHRONOUS EVENTS

34453        Default.

### 34454 STDOUT

34455        If standard output is a terminal, the appropriate sequence to clear and set the tab stops may be written to standard output in an unspecified format. If standard output is not a terminal, undefined results occur.

### 34458 STDERR

34459        The standard error shall be used only for diagnostic messages.

### 34460 OUTPUT FILES

34461        None.

## 34462 EXTENDED DESCRIPTION

34463 None.

## 34464 EXIT STATUS

34465 The following exit values shall be returned:

34466 0 Successful completion.

34467 >0 An error occurred.

## 34468 CONSEQUENCES OF ERRORS

34469 Default.

## 34470 APPLICATION USAGE

34471 This utility makes use of the terminal's hardware tabs and the *stty tabs* option.

34472 This utility is not recommended for application use.

34473 Some integrated display units might not have escape sequences to set tab stops, but may be set  
34474 by internal system calls. On these terminals, *tabs* works if standard output is directed to the  
34475 terminal; if output is directed to another file, however, *tabs* fails.

## 34476 EXAMPLES

34477 None.

## 34478 RATIONALE

34479 Consideration was given to having the *tput* utility handle all of the functions described in *tabs*.  
34480 However, the separate *tabs* utility was retained because it seems more intuitive to use a  
34481 command named *tabs* than *tput* with a new option. The *tput* utility does not support setting or  
34482 clearing tabs, and no known historical version of *tabs* supports the capability of setting arbitrary  
34483 tab stops.

34484 The System V *tabs* interface is very complex; the version in this volume of IEEE Std 1003.1-2001  
34485 has a reduced feature list, but many of the features omitted were restored as XSI extensions even  
34486 though the supported languages and coding styles are primarily historical.

34487 There was considerable sentiment for specifying only a means of resetting the tabs back to a  
34488 known state—presumably the “standard” of tabs every eight positions. The following features  
34489 were omitted:

- 34490 • Setting tab stops via the first line in a file, using *--file*. Since even the SVID has no complete  
34491 explanation of this feature, it is doubtful that it is in widespread use.

34492 In an early proposal, a *-t tablist* option was added for consistency with *expand*; this was later  
34493 removed when inconsistencies with the historical list of tabs were identified.

34494 Consideration was given to adding a *-p* option that would output the current tab settings so  
34495 that they could be saved and then later restored. This was not accepted because querying the tab  
34496 stops of the terminal is not a capability in historical *terminfo* or *termcap* facilities and might not be  
34497 supported on a wide range of terminals.

## 34498 FUTURE DIRECTIONS

34499 None.

## 34500 SEE ALSO

34501 *expand*, *stty*, *tput*, *unexpand*

**34502 CHANGE HISTORY**

34503 First released in Issue 2.

**34504 Issue 6**

34505 This utility is marked as part of the User Portability Utilities option.

34506 The normative text is reworded to avoid use of the term “must” for application requirements.

## 34507 NAME

34508        tail — copy the last part of a file

## 34509 SYNOPSIS

34510        tail [-f][ -c number| -n number][file]

## 34511 DESCRIPTION

34512        The *tail* utility shall copy its input file to the standard output beginning at a designated place.

34513        Copying shall begin at the point in the file indicated by the **-c** *number* or **-n** *number* options. The option-argument *number* shall be counted in units of lines or bytes, according to the options **-n** and **-c**. Both line and byte counts start from 1.

34516        Tails relative to the end of the file may be saved in an internal buffer, and thus may be limited in length. Such a buffer, if any, shall be no smaller than {LINE\_MAX}\*10 bytes.

## 34518 OPTIONS

34519        The *tail* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

34521        The following options shall be supported:

34522        **-c** *number*    The application shall ensure that the *number* option-argument is a decimal integer whose sign affects the location in the file, measured in bytes, to begin the copying:

| Sign | Copying Starts                         |
|------|----------------------------------------|
| +    | Relative to the beginning of the file. |
| -    | Relative to the end of the file.       |
| none | Relative to the end of the file.       |

34528        The origin for counting shall be 1; that is, **-c +1** represents the first byte of the file, **-c -1** the last.

34530        **-f**        If the input file is a regular file or if the *file* operand specifies a FIFO, do not terminate after the last line of the input file has been copied, but read and copy further bytes from the input file when they become available. If no *file* operand is specified and standard input is a pipe, the **-f** option shall be ignored. If the input file is not a FIFO, pipe, or regular file, it is unspecified whether or not the **-f** option shall be ignored.

34536        **-n** *number*    This option shall be equivalent to **-c** *number*, except the starting location in the file shall be measured in lines instead of bytes. The origin for counting shall be 1; that is, **-n +1** represents the first line of the file, **-n -1** the last.

34539        If neither **-c** nor **-n** is specified, **-n 10** shall be assumed.

## 34540 OPERANDS

34541        The following operand shall be supported:

34542        *file*        A pathname of an input file. If no *file* operands are specified, the standard input shall be used.

## 34544 STDIN

34545        The standard input shall be used only if no *file* operands are specified. See the INPUT FILES section.

**34547 INPUT FILES**

34548     If the **-c** option is specified, the input file can contain arbitrary data; otherwise, the input file  
34549     shall be a text file.

**34550 ENVIRONMENT VARIABLES**

34551     The following environment variables shall affect the execution of *tail*:

34552     **LANG**     Provide a default value for the internationalization variables that are unset or null.  
34553         (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
34554         Internationalization Variables for the precedence of internationalization variables  
34555         used to determine the values of locale categories.)

34556     **LC\_ALL**     If set to a non-empty string value, override the values of all the other  
34557         internationalization variables.

34558     **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
34559         characters (for example, single-byte as opposed to multi-byte characters in  
34560         arguments and input files).

**34561 LC\_MESSAGES**

34562         Determine the locale that should be used to affect the format and contents of  
34563         diagnostic messages written to standard error.

34564 XSI     **NLSPATH**     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**34565 ASYNCHRONOUS EVENTS**

34566         Default.

**34567 STDOUT**

34568         The designated portion of the input file shall be written to standard output.

**34569 STDERR**

34570         The standard error shall be used only for diagnostic messages.

**34571 OUTPUT FILES**

34572         None.

**34573 EXTENDED DESCRIPTION**

34574         None.

**34575 EXIT STATUS**

34576         The following exit values shall be returned:

34577            0     Successful completion.

34578            >0    An error occurred.

**34579 CONSEQUENCES OF ERRORS**

34580         Default.

**34581 APPLICATION USAGE**

34582         The **-c** option should be used with caution when the input is a text file containing multi-byte  
34583         characters; it may produce output that does not start on a character boundary.

34584         Although the input file to *tail* can be any type, the results might not be what would be expected  
34585         on some character special device files or on file types not described by the System Interfaces  
34586         volume of IEEE Std 1003.1-2001. Since this volume of IEEE Std 1003.1-2001 does not specify the  
34587         block size used when doing input, *tail* need not read all of the data from devices that only  
34588         perform block transfers.

**34589 EXAMPLES**

34590     The **-f** option can be used to monitor the growth of a file that is being written by some other  
34591     process. For example, the command:

34592       tail -f fred

34593     prints the last ten lines of the file **fred**, followed by any lines that are appended to **fred** between  
34594     the time **tail** is initiated and killed. As another example, the command:

34595       tail -f -c 15 fred

34596     prints the last 15 bytes of the file **fred**, followed by any bytes that are appended to **fred** between  
34597     the time **tail** is initiated and killed.

**34598 RATIONALE**

34599     This version of **tail** was created to allow conformance to the Utility Syntax Guidelines. The  
34600     historical **-b** option was omitted because of the general non-portability of block-sized units of  
34601     text. The **-c** option historically meant “characters”, but this volume of IEEE Std 1003.1-2001  
34602     indicates that it means “bytes”. This was selected to allow reasonable implementations when  
34603     multi-byte characters are possible; it was not named **-b** to avoid confusion with the historical  
34604     **-b**.

34605     The origin of counting both lines and bytes is 1, matching all widespread historical  
34606     implementations.

34607     The restriction on the internal buffer is a compromise between the historical System V  
34608     implementation of 4 096 bytes and the BSD 32 768 bytes.

34609     The **-f** option has been implemented as a loop that sleeps for 1 second and copies any bytes that  
34610     are available. This is sufficient, but if more efficient methods of determining when new data are  
34611     available are developed, implementations are encouraged to use them.

34612     Historical documentation indicates that **tail** ignores the **-f** option if the input file is a pipe (pipe  
34613     and FIFO on systems that support FIFOs). On BSD-based systems, this has been true; on System  
34614     V-based systems, this was true when input was taken from standard input, but it did not ignore  
34615     the **-f** flag if a FIFO was named as the *file* operand. Since the **-f** option is not useful on pipes and  
34616     all historical implementations ignore **-f** if no *file* operand is specified and standard input is a  
34617     pipe, this volume of IEEE Std 1003.1-2001 requires this behavior. However, since the **-f** option is  
34618     useful on a FIFO, this volume of IEEE Std 1003.1-2001 also requires that if standard input is a  
34619     FIFO or a FIFO is named, the **-f** option shall not be ignored. Although historical behavior does  
34620     not ignore the **-f** option for other file types, this is unspecified so that implementations are  
34621     allowed to ignore the **-f** option if it is known that the file cannot be extended.

34622     This was changed to the current form based on comments noting that **-c** was almost never used  
34623     without specifying a number and that there was no need to specify **-l** if **-n number** was given.

**34624 FUTURE DIRECTIONS**

34625     None.

**34626 SEE ALSO**

34627       **head**

**34628 CHANGE HISTORY**

34629     First released in Issue 2.

**34630 Issue 6**

34631     The obsolescent SYNOPSIS lines and associated text are removed.

34632     The normative text is reworded to avoid use of the term “must” for application requirements.

## 34633 NAME

34634 talk — talk to another user

## 34635 SYNOPSIS

34636 UP talk address [terminal]

34637

## 34638 DESCRIPTION

34639 The *talk* utility is a two-way, screen-oriented communication program.34640 When first invoked, *talk* shall send a message similar to:34641 Message from <unspecified string>  
34642 talk: connection requested by *your\_address*  
34643 talk: respond with: talk *your\_address*34644 to the specified *address*. At this point, the recipient of the message can reply by typing:34645 talk *your\_address*34646 Once communication is established, the two parties can type simultaneously, with their output  
34647 displayed in separate regions of the screen. Characters shall be processed as follows:

- 34648 • Typing the alert character shall alert the recipient's terminal.
- 34649 • Typing <control>-L shall cause the sender's screen regions to be refreshed.
- 34650 • Typing the erase and kill characters shall affect the sender's terminal in the manner described  
34651 by the **termios** interface in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11,  
34652 General Terminal Interface.
- 34653 • Typing the interrupt or end-of-file characters shall terminate the local *talk* utility. Once the  
34654 session has been terminated on one side, the other side of the *talk* session shall be notified  
34655 that the *talk* session has been terminated and shall be able to do nothing except exit.
- 34656 • Typing characters from *LC\_CTYPE* classifications **print** or **space** shall cause those characters  
34657 to be sent to the recipient's terminal.
- 34658 • When and only when the *stty iexten* local mode is enabled, the existence and processing of  
34659 additional special control characters and multi-byte or single-byte functions shall be  
34660 implementation-defined.
- 34661 • Typing other non-printable characters shall cause implementation-defined sequences of  
34662 printable characters to be sent to the recipient's terminal.

34663 Permission to be a recipient of a *talk* message can be denied or granted by use of the *mesg* utility.  
34664 However, a user's privilege may further constrain the domain of accessibility of other users'  
34665 terminals. The *talk* utility shall fail when the user lacks the appropriate privileges to perform the  
34666 requested action.34667 Certain block-mode terminals do not have all the capabilities necessary to support the  
34668 simultaneous exchange of messages required for *talk*. When this type of exchange cannot be  
34669 supported on such terminals, the implementation may support an exchange with reduced levels  
34670 of simultaneous interaction or it may report an error describing the terminal-related deficiency.

## 34671 OPTIONS

34672 None.

## 34673 OPERANDS

34674 The following operands shall be supported:

34675 **address** The recipient of the *talk* session. One form of *address* is the <*user name*>, as returned by the *who* utility. Other address formats and how they are handled are unspecified.

34678 **terminal** If the recipient is logged in more than once, the *terminal* argument can be used to indicate the appropriate terminal name. If *terminal* is not specified, the *talk* message shall be displayed on one or more accessible terminals in use by the recipient. The format of *terminal* shall be the same as that returned by the *who* utility.

## 34682 STDIN

34683 Characters read from standard input shall be copied to the recipient's terminal in an unspecified manner. If standard input is not a terminal, *talk* shall write a diagnostic message and exit with a non-zero status.

## 34686 INPUT FILES

34687 None.

## 34688 ENVIRONMENT VARIABLES

34689 The following environment variables shall affect the execution of *talk*:

34690 **LANG** Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

34694 **LC\_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

34696 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files). If the recipient's locale does not use an *LC\_CTYPE* equivalent to the sender's, the results are undefined.

34700 **LC\_MESSAGES**

34701 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

34704 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

34705 **TERM** Determine the name of the invoker's terminal type. If this variable is unset or null, an unspecified default terminal type shall be used.

## 34707 ASYNCHRONOUS EVENTS

34708 When the *talk* utility receives a SIGINT signal, the utility shall terminate and exit with a zero status. It shall take the standard action for all other signals.

## 34710 STDOUT

34711 If standard output is a terminal, characters copied from the recipient's standard input may be written to standard output. Standard output also may be used for diagnostic messages. If standard output is not a terminal, *talk* shall exit with a non-zero status.

## 34714 STDERR

34715 None.

**34716 OUTPUT FILES**

34717 None.

**34718 EXTENDED DESCRIPTION**

34719 None.

**34720 EXIT STATUS**

34721 The following exit values shall be returned:

34722 0 Successful completion.

34723 >0 An error occurred or *talk* was invoked on a terminal incapable of supporting it.

**34724 CONSEQUENCES OF ERRORS**

34725 Default.

**34726 APPLICATION USAGE**

34727 Because the handling of non-printable, non-<space>s is tied to the *stty* description of **iexten**,  
34728 implementation extensions within the terminal driver can be accessed. For example, some  
34729 implementations provide line editing functions with certain control character sequences.

**34730 EXAMPLES**

34731 None.

**34732 RATIONALE**

34733 The *write* utility was included in this volume of IEEE Std 1003.1-2001 since it can be  
34734 implemented on all terminal types. The *talk* utility, which cannot be implemented on certain  
34735 terminals, was considered to be a “better” communications interface. Both of these programs are  
34736 in widespread use on historical implementations. Therefore, both utilities have been specified.

34737 All references to networking abilities (*talking* to a user on another system) were removed as  
34738 being outside the scope of this volume of IEEE Std 1003.1-2001.

34739 Historical BSD and System V versions of *talk* terminate both of the conversations when either  
34740 user breaks out of the session. This can lead to adverse consequences if a user unwittingly  
34741 continues to enter text that is interpreted by the shell when the other terminates the session.  
34742 Therefore, the version of *talk* specified by this volume of IEEE Std 1003.1-2001 requires both  
34743 users to terminate their end of the session explicitly.

34744 Only messages sent to the terminal of the invoking user can be internationalized in any way:

- 34745 • The original “Message from *<unspecified string>* ...” message sent to the terminal of the  
34746 recipient cannot be internationalized because the environment of the recipient is as yet  
34747 inaccessible to the *talk* utility. The environment of the invoking party is irrelevant.
- 34748 • Subsequent communication between the two parties cannot be internationalized because the  
34749 two parties may specify different languages in their environment (and non-portable  
34750 characters cannot be mapped from one language to another).
- 34751 • Neither party can be required to communicate in a language other than C and/or the one  
34752 specified by their environment because unavailable terminal hardware support (for example,  
34753 fonts) may be required.

34754 The text in the STDOOUT section reflects the usage of the verb “display” in this section; some *talk*  
34755 implementations actually use standard output to write to the terminal, but this volume of  
34756 IEEE Std 1003.1-2001 does not require that to be the case.

34757 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
34758 require that they all use or accept the same format.

34759        The handling of non-printable characters is partially implementation-defined because the details  
34760        of mapping them to printable sequences is not needed by the user. Historical implementations,  
34761        for security reasons, disallow the transmission of non-printable characters that may send  
34762        commands to the other terminal.

34763 **FUTURE DIRECTIONS**

34764        None.

34765 **SEE ALSO**

34766        *mesg*, *stty*, *who*, *write*, the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General  
34767        Terminal Interface

34768 **CHANGE HISTORY**

34769        First released in Issue 4.

34770 **Issue 6**

34771        This utility is marked as part of the User Portability Utilities option.

**34772 NAME**

34773        tee — duplicate standard input

**34774 SYNOPSIS**

34775        tee [-ai][*file...*]

**34776 DESCRIPTION**

34777        The *tee* utility shall copy standard input to standard output, making a copy in zero or more files.

34778        The *tee* utility shall not buffer output.

34779        If the **-a** option is not specified, output files shall be written (see Section 1.7.1.4 (on page 4)).

**34780 OPTIONS**

34781        The *tee* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

34783        The following options shall be supported:

34784        **-a**        Append the output to the files.

34785        **-i**        Ignore the SIGINT signal.

**34786 OPERANDS**

34787        The following operands shall be supported:

34788        *file*        A pathname of an output file. Processing of at least 13 *file* operands shall be supported.

**34790 STDIN**

34791        The standard input can be of any type.

**34792 INPUT FILES**

34793        None.

**34794 ENVIRONMENT VARIABLES**

34795        The following environment variables shall affect the execution of *tee*:

34796        **LANG**      Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

34800        **LC\_ALL**     If set to a non-empty string value, override the values of all the other internationalization variables.

34802        **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

**34805 *LC\_MESSAGES***

34806        Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

34808 XSI      **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**34809 ASYNCHRONOUS EVENTS**

34810        Default, except that if the **-i** option was specified, SIGINT shall be ignored.

**34811 STDOUT**

34812 The standard output shall be a copy of the standard input.

**34813 STDERR**

34814 The standard error shall be used only for diagnostic messages.

**34815 OUTPUT FILES**

34816 If any *file* operands are specified, the standard input shall be copied to each named file.

**34817 EXTENDED DESCRIPTION**

34818 None.

**34819 EXIT STATUS**

34820 The following exit values shall be returned:

34821 0 The standard input was successfully copied to all output files.

34822 >0 An error occurred.

**34823 CONSEQUENCES OF ERRORS**

34824 If a write to any successfully opened *file* operand fails, writes to other successfully opened *file* operands and standard output shall continue, but the exit status shall be non-zero. Otherwise, the default actions specified in Section 1.11 (on page 20) apply.

**34827 APPLICATION USAGE**

34828 The *tee* utility is usually used in a pipeline, to make a copy of the output of some utility.

34829 The *file* operand is technically optional, but *tee* is no more useful than *cat* when none is specified.

**34830 EXAMPLES**

34831 Save an unsorted intermediate form of the data in a pipeline:

34832 ... | tee unsorted | sort > sorted

**34833 RATIONALE**

34834 The buffering requirement means that *tee* is not allowed to use ISO C standard fully buffered or line-buffered writes. It does not mean that *tee* has to do 1-byte reads followed by 1-byte writes.

34836 It should be noted that early versions of BSD ignore any invalid options and accept a single '-' as an alternative to -i. They also print a message if unable to open a file:

34838 "tee: cannot access %s\n", <pathname>

34839 Historical implementations ignore write errors. This is explicitly not permitted by this volume of IEEE Std 1003.1-2001.

34841 Some historical implementations use O\_APPEND when providing append mode; others use the lseek() function to seek to the end-of-file after opening the file without O\_APPEND. This volume of IEEE Std 1003.1-2001 requires functionality equivalent to using O\_APPEND; see Section 1.7.1.4 (on page 4).

**34845 FUTURE DIRECTIONS**

34846 None.

**34847 SEE ALSO**

34848 Chapter 1 (on page 1), *cat*, the System Interfaces volume of IEEE Std 1003.1-2001, *lseek()*

**34849 CHANGE HISTORY**

34850 First released in Issue 2.

34851 **Issue 6**

34852 IEEE PASC Interpretation 1003.2 #168 is applied.

## 34853 NAME

34854 test — evaluate expression

## 34855 SYNOPSIS

34856 test [*expression*]

34857 [ [*expression*] ]

## 34858 DESCRIPTION

34859 The *test* utility shall evaluate the *expression* and indicate the result of the evaluation by its exit status. An exit status of zero indicates that the expression evaluated as true and an exit status of 1 indicates that the expression evaluated as false.

34862 In the second form of the utility, which uses "[]" rather than *test*, the application shall ensure  
34863 that the square brackets are separate arguments.

## 34864 OPTIONS

34865 The *test* utility shall not recognize the "--" argument in the manner specified by guideline 10 in  
34866 the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

34867 No options shall be supported.

## 34868 OPERANDS

34869 The application shall ensure that all operators and elements of primaries are presented as  
34870 separate arguments to the *test* utility.

34871 The following primaries can be used to construct *expression*:

34872 **-b** *file* True if *file* exists and is a block special file.

34873 **-c** *file* True if *file* exists and is a character special file.

34874 **-d** *file* True if *file* exists and is a directory.

34875 **-e** *file* True if *file* exists.

34876 **-f** *file* True if *file* exists and is a regular file.

34877 **-g** *file* True if *file* exists and its set-group-ID flag is set.

34878 **-h** *file* True if *file* exists and is a symbolic link.

34879 **-L** *file* True if *file* exists and is a symbolic link.

34880 **-n** *string* True if the length of *string* is non-zero.

34881 **-p** *file* True if *file* is a FIFO.

34882 **-r** *file* True if *file* exists and is readable. True shall indicate that permission to read from  
34883 *file* will be granted, as defined in Section 1.7.1.4 (on page 4).

34884 **-S** *file* True if *file* exists and is a socket.

34885 **-s** *file* True if *file* exists and has a size greater than zero.

34886 **-t** *file\_descriptor*

34887 True if the file whose file descriptor number is *file\_descriptor* is open and is  
34888 associated with a terminal.

34889 **-u** *file* True if *file* exists and its set-user-ID flag is set.

34890 **-w** *file* True if *file* exists and is writable. True shall indicate that permission to write from  
34891 *file* will be granted, as defined in Section 1.7.1.4 (on page 4).

|           |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34892     | <b>-x file</b>                                                                                                                                                                                                                                                                                 | True if <i>file</i> exists and is executable. True shall indicate that permission to execute <i>file</i> will be granted, as defined in Section 1.7.1.4 (on page 4). If <i>file</i> is a directory, true shall indicate that permission to search <i>file</i> will be granted. |
| 34893     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34894     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34895     | <b>-z string</b>                                                                                                                                                                                                                                                                               | True if the length of string <i>string</i> is zero.                                                                                                                                                                                                                            |
| 34896     | <i>string</i>                                                                                                                                                                                                                                                                                  | True if the string <i>string</i> is not the null string.                                                                                                                                                                                                                       |
| 34897     | <i>s1 = s2</i>                                                                                                                                                                                                                                                                                 | True if the strings <i>s1</i> and <i>s2</i> are identical.                                                                                                                                                                                                                     |
| 34898     | <i>s1 != s2</i>                                                                                                                                                                                                                                                                                | True if the strings <i>s1</i> and <i>s2</i> are not identical.                                                                                                                                                                                                                 |
| 34899     | <i>n1 -eq n2</i>                                                                                                                                                                                                                                                                               | True if the integers <i>n1</i> and <i>n2</i> are algebraically equal.                                                                                                                                                                                                          |
| 34900     | <i>n1 -ne n2</i>                                                                                                                                                                                                                                                                               | True if the integers <i>n1</i> and <i>n2</i> are not algebraically equal.                                                                                                                                                                                                      |
| 34901     | <i>n1 -gt n2</i>                                                                                                                                                                                                                                                                               | True if the integer <i>n1</i> is algebraically greater than the integer <i>n2</i> .                                                                                                                                                                                            |
| 34902     | <i>n1 -ge n2</i>                                                                                                                                                                                                                                                                               | True if the integer <i>n1</i> is algebraically greater than or equal to the integer <i>n2</i> .                                                                                                                                                                                |
| 34903     | <i>n1 -lt n2</i>                                                                                                                                                                                                                                                                               | True if the integer <i>n1</i> is algebraically less than the integer <i>n2</i> .                                                                                                                                                                                               |
| 34904     | <i>n1 -le n2</i>                                                                                                                                                                                                                                                                               | True if the integer <i>n1</i> is algebraically less than or equal to the integer <i>n2</i> .                                                                                                                                                                                   |
| 34905 XSI | <b>expression1 -a expression2</b>                                                                                                                                                                                                                                                              | True if both <i>expression1</i> and <i>expression2</i> are true. The <b>-a</b> binary primary is left associative. It has a higher precedence than <b>-o</b> .                                                                                                                 |
| 34906     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34907     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34908 XSI | <b>expression1 -o expression2</b>                                                                                                                                                                                                                                                              | True if either <i>expression1</i> or <i>expression2</i> is true. The <b>-o</b> binary primary is left associative.                                                                                                                                                             |
| 34909     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34910     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34911     | With the exception of the <b>-h file</b> and <b>-L file</b> primaries, if a <i>file</i> argument is a symbolic link, <i>test</i> shall evaluate the expression by resolving the symbolic link and using the file referenced by the link.                                                       |                                                                                                                                                                                                                                                                                |
| 34912     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34913     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34914     | These primaries can be combined with the following operators:                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                |
| 34915     | <b>! expression</b>                                                                                                                                                                                                                                                                            | True if <i>expression</i> is false.                                                                                                                                                                                                                                            |
| 34916 XSI | <b>( expression )</b>                                                                                                                                                                                                                                                                          | True if <i>expression</i> is true. The parentheses can be used to alter the normal precedence and associativity.                                                                                                                                                               |
| 34917     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34918     | The primaries with two elements of the form:                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                |
| 34919     | <i>-primary_operator primary_operand</i>                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                |
| 34920     | are known as <i>unary primaries</i> . The primaries with three elements in either of the two forms:                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                |
| 34921     | <i>primary_operand -primary_operator primary_operand</i>                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                |
| 34922     | <i>primary_operand primary_operator primary_operand</i>                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                |
| 34923     | are known as <i>binary primaries</i> . Additional implementation-defined operators and <i>primary_operators</i> may be provided by implementations. They shall be of the form <b>-operator</b> where the first character of <i>operator</i> is not a digit.                                    |                                                                                                                                                                                                                                                                                |
| 34924     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34925     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34926     | The algorithm for determining the precedence of the operators and the return value that shall be generated is based on the number of arguments presented to <i>test</i> . (However, when using the "[ . . . ]" form, the right-bracket final argument shall not be counted in this algorithm.) |                                                                                                                                                                                                                                                                                |
| 34927     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34928     |                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                |
| 34929     | In the following list, \$1, \$2, \$3, and \$4 represent the arguments presented to <i>test</i> :                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                |
| 34930     | 0 arguments:                                                                                                                                                                                                                                                                                   | Exit false (1).                                                                                                                                                                                                                                                                |

|           |               |                                                                                                                                                          |
|-----------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34931     | 1 argument:   | Exit true (0) if \$1 is not null; otherwise, exit false.                                                                                                 |
| 34932     | 2 arguments:  | <ul style="list-style-type: none"><li>If \$1 is '!', exit true if \$2 is null, false if \$2 is not null.</li></ul>                                       |
| 34933     |               | <ul style="list-style-type: none"><li>If \$1 is a unary primary, exit true if the unary test is true, false if the unary test is false.</li></ul>        |
| 34934     |               | <ul style="list-style-type: none"><li>Otherwise, produce unspecified results.</li></ul>                                                                  |
| 34935     |               |                                                                                                                                                          |
| 34936     | 3 arguments:  | <ul style="list-style-type: none"><li>If \$2 is a binary primary, perform the binary test of \$1 and \$3.</li></ul>                                      |
| 34937     |               | <ul style="list-style-type: none"><li>If \$1 is '!', negate the two-argument test of \$2 and \$3.</li></ul>                                              |
| 34938     |               | <ul style="list-style-type: none"><li>If \$1 is '()' and \$3 is ')', perform the unary test of \$2.</li></ul>                                            |
| 34939     |               | <ul style="list-style-type: none"><li>Otherwise, produce unspecified results.</li></ul>                                                                  |
| 34940     | 4 arguments:  | <ul style="list-style-type: none"><li>If \$1 is '!', negate the three-argument test of \$2, \$3, and \$4.</li></ul>                                      |
| 34941 XSI |               | <ul style="list-style-type: none"><li>If \$1 is '()' and \$4 is ')', perform the two-argument test of \$2 and \$3.</li></ul>                             |
| 34942     |               | <ul style="list-style-type: none"><li>Otherwise, the results are unspecified.</li></ul>                                                                  |
| 34943     | >4 arguments: | The results are unspecified.                                                                                                                             |
| 34944 XSI |               | On XSI-conformant systems, combinations of primaries and operators shall be evaluated using the precedence and associativity rules described previously. |
| 34945     |               | In addition, the string comparison binary primaries '=' and "!=" shall have a higher precedence than any unary primary.                                  |
| 34946     |               |                                                                                                                                                          |
| 34947     |               |                                                                                                                                                          |

#### 34948 STDIN

34949 Not used.

#### 34950 INPUT FILES

34951 None.

#### 34952 ENVIRONMENT VARIABLES

34953 The following environment variables shall affect the execution of *test*:

34954 *LANG* Provide a default value for the internationalization variables that are unset or null.  
34955 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
34956 Internationalization Variables for the precedence of internationalization variables  
34957 used to determine the values of locale categories.)

34958 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
34959 internationalization variables.

34960 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
34961 characters (for example, single-byte as opposed to multi-byte characters in  
34962 arguments).

34963 *LC\_MESSAGES* Determine the locale that should be used to affect the format and contents of  
34964 diagnostic messages written to standard error.

34966 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

#### 34967 ASYNCHRONOUS EVENTS

34968 Default.

**34969 STDOUT**

34970 Not used.

**34971 STDERR**

34972 The standard error shall be used only for diagnostic messages.

**34973 OUTPUT FILES**

34974 None.

**34975 EXTENDED DESCRIPTION**

34976 None.

**34977 EXIT STATUS**

34978 The following exit values shall be returned:

34979 0 *expression* evaluated to true.

34980 1 *expression* evaluated to false or *expression* was missing.

34981 >1 An error occurred.

**34982 CONSEQUENCES OF ERRORS**

34983 Default.

**34984 APPLICATION USAGE**

34985 Scripts should be careful when dealing with user-supplied input that could be confused with  
34986 primaries and operators. Unless the application writer knows all the cases that produce input to  
34987 the script, invocations like:

34988 `test "$1" -a "$2"`

34989 should be written as:

34990 `test "$1" && test "$2"`

34991 to avoid problems if a user supplied values such as \$1 set to '!' and \$2 set to the null string.  
34992 That is, in cases where maximal portability is of concern, replace:

34993 `test expr1 -a expr2`

34994 with:

34995 `test expr1 && test expr2`

34996 and replace:

34997 `test expr1 -o expr2`

34998 with:

34999 `test expr1 || test expr2`

35000 but note that, in `test`, `-a` has higher precedence than `-o` while "`&&`" and "`||`" have equal  
35001 precedence in the shell.

35002 Parentheses or braces can be used in the shell command language to effect grouping.

35003 Parentheses must be escaped when using `sh`; for example:

35004 `test \(\ expr1 -a expr2 \) -o expr3`

35005 This command is not always portable outside XSI-conformant systems. The following form can  
35006 be used instead:

```
35007 (test expr1 && test expr2) || test expr3
```

35008 The two commands:

```
35009 test "$1"
35010 test ! "$1"
```

35011 could not be used reliably on some historical systems. Unexpected results would occur if such a  
35012 *string* expression were used and \$1 expanded to '!', '(', or a known unary primary. Better  
35013 constructs are:

```
35014 test -n "$1"
35015 test -z "$1"
```

35016 respectively.

35017 Historical systems have also been unreliable given the common construct:

```
35018 test "$response" = "expected string"
```

35019 One of the following is a more reliable form:

```
35020 test "X$response" = "Xexpected string"
35021 test "expected string" = "$response"
```

35022 Note that the second form assumes that *expected string* could not be confused with any unary  
35023 primary. If *expected string* starts with '−', '(', ')', or even '=' , the first form should be used  
35024 instead. Using the preceding rules without the XSI marked extensions, any of the three  
35025 comparison forms is reliable, given any input. (However, note that the strings are quoted in all  
35026 cases.)

35027 Because the string comparison binary primaries, '=' and "!=" , have a higher precedence than  
35028 any unary primary in the greater than 4 argument case, unexpected results can occur if  
35029 arguments are not properly prepared. For example, in:

```
35030 test -d $1 -o -d $2
```

35031 If \$1 evaluates to a possible directory name of '=' , the first three arguments are considered a  
35032 string comparison, which shall cause a syntax error when the second -d is encountered. One of  
35033 the following forms prevents this; the second is preferred:

```
35034 test \(-d "$1" \) -o \(-d "$2" \)
35035 test -d "$1" || test -d "$2"
```

35036 Also in the greater than 4 argument case:

```
35037 test "$1" = "bat" -a "$2" = "ball"
```

35038 syntax errors occur if \$1 evaluates to '(' or ')'. One of the following forms prevents this; the  
35039 third is preferred:

```
35040 test "X$1" = "Xbat" -a "X$2" = "Xball"
35041 test "$1" = "bat" && test "$2" = "ball"
35042 test "X$1" = "Xbat" && test "X$2" = "Xball"
```

## 35043 EXAMPLES

35044 1. Exit if there are not two or three arguments (two variations):

```
35045 if [$# -ne 2 -a $# -ne 3]; then exit 1; fi
35046 if [$# -lt 2 -o $# -gt 3]; then exit 1; fi
```

```

35047 2. Perform a mkdir if a directory does not exist:
35048 test ! -d tempdir && mkdir tempdir
35049 3. Wait for a file to become non-readable:
35050 while test -r thefile
35051 do
35052 sleep 30
35053 done
35054 echo '"thefile" is no longer readable'
35055 4. Perform a command if the argument is one of three strings (two variations):
35056 if ["$1" = "pear"] || ["$1" = "grape"] || ["$1" = "apple"]
35057 then
35058 command
35059 fi
35060 case "$1" in
35061 pear|grape|apple) command ;;
35062 esac

```

### 35063 RATIONALE

35064 The KornShell-derived conditional command (double bracket [[]]) was removed from the shell  
 35065 command language description in an early proposal. Objections were raised that the real  
 35066 problem is misuse of the *test* command (!), and putting it into the shell is the wrong way to fix  
 35067 the problem. Instead, proper documentation and a new shell reserved word (!) are sufficient.

35068 Tests that require multiple *test* operations can be done at the shell level using individual  
 35069 invocations of the *test* command and shell logicals, rather than using the error-prone **-o** flag of  
 35070 *test*.

35071 XSI-conformant systems support more than four arguments.

35072 XSI-conformant systems support the combining of primaries with the following constructs:

35073 ***expression1 -a expression2***

35074 True if both *expression1* and *expression2* are true.

35075 ***expression1 -o expression2***

35076 True if at least one of *expression1* and *expression2* are true.

35077 **( *expression* )**

35078 True if *expression* is true.

35079 In evaluating these more complex combined expressions, the following precedence rules are  
 35080 used:

- 35081 • The unary primaries have higher precedence than the algebraic binary primaries.
- 35082 • The unary primaries have lower precedence than the string binary primaries.
- 35083 • The unary and binary primaries have higher precedence than the unary *string* primary.
- 35084 • The **!** operator has higher precedence than the **-a** operator, and the **-a** operator has higher  
 35085 precedence than the **-o** operator.
- 35086 • The **-a** and **-o** operators are left associative.
- 35087 • The parentheses can be used to alter the normal precedence and associativity.

35088        The BSD and System V versions of **-f** are not the same. The BSD definition was:  
35089        **-f file**      True if *file* exists and is not a directory.  
35090        The SVID version (true if the file exists and is a regular file) was chosen for this volume of  
35091        IEEE Std 1003.1-2001 because its use is consistent with the **-b**, **-c**, **-d**, and **-p** operands (*file* exists  
35092        and is a specific file type).  
35093        The **-e** primary, possessing similar functionality to that provided by the C shell, was added  
35094        because it provides the only way for a shell script to find out if a file exists without trying to  
35095        open the file. Since implementations are allowed to add additional file types, a portable script  
35096        cannot use:  
35097        

```
test -b foo -o -c foo -o -d foo -o -f foo -o -p foo
```

  
35098        to find out if **foo** is an existing file. On historical BSD systems, the existence of a file could be  
35099        determined by:  
35100        

```
test -f foo -o -d foo
```

  
35101        but there was no easy way to determine that an existing file was a regular file. An early proposal  
35102        used the KornShell **-a** primary (with the same meaning), but this was changed to **-e** because  
35103        there were concerns about the high probability of humans confusing the **-a** primary with the **-a**  
35104        binary operator.  
35105        The following options were not included in this volume of IEEE Std 1003.1-2001, although they  
35106        are provided by some implementations. These operands should not be used by new  
35107        implementations for other purposes:  
35108        **-k file**      True if *file* exists and its sticky bit is set.  
35109        **-C file**      True if *file* is a contiguous file.  
35110        **-V file**      True if *file* is a version file.  
35111        The following option was not included because it was undocumented in most implementations,  
35112        has been removed from some implementations (including System V), and the functionality is  
35113        provided by the shell (see Section 2.6.2 (on page 37)).  
35114        **-l string**     The length of the string *string*.  
35115        The **-b**, **-c**, **-g**, **-p**, **-u**, and **-x** operands are derived from the SVID; historical BSD does not  
35116        provide them. The **-k** operand is derived from System V; historical BSD does not provide it.  
35117        On historical BSD systems, *test -w directory* always returned false because *test* tried to open the  
35118        directory for writing, which always fails.  
35119        Some additional primaries newly invented or from the KornShell appeared in an early proposal  
35120        as part of the conditional command ([[]]): *s1 > s2*, *s1 < s2*, *str = pattern*, *str != pattern*, *f1 -nt f2*, *f1*  
35121        **-ot f2**, and *f1 -ef f2*. They were not carried forward into the *test* utility when the conditional  
35122        command was removed from the shell because they have not been included in the *test* utility  
35123        built into historical implementations of the *sh* utility.  
35124        The **-t file\_descriptor** primary is shown with a mandatory argument because the grammar is  
35125        ambiguous if it can be omitted. Historical implementations have allowed it to be omitted,  
35126        providing a default of 1.  
35127        **FUTURE DIRECTIONS**  
35128        None.

**35129 SEE ALSO**

35130       Section 1.7.1.4 (on page 4), *find*

**35131 CHANGE HISTORY**

35132       First released in Issue 2.

**35133 Issue 5**

35134       The FUTURE DIRECTIONS section is added.

**35135 Issue 6**

35136       The **-h** operand is added for symbolic links, and access permission requirements are clarified for  
35137       the **-r**, **-w**, and **-x** operands to align with the IEEE P1003.2b draft standard.

35138       The normative text is reworded to avoid use of the term “must” for application requirements.

35139       The **-L** and **-S** operands are added for symbolic links and sockets.

## 35140 NAME

35141 time — time a simple command

## 35142 SYNOPSIS

35143 UP time [-p] utility [argument...]

35144

## 35145 DESCRIPTION

35146 The *time* utility shall invoke the utility named by the *utility* operand with arguments supplied as  
35147 the *argument* operands and write a message to standard error that lists timing statistics for the  
35148 utility. The message shall include the following information:

- 35149 • The elapsed (real) time between invocation of *utility* and its termination.
- 35150 • The User CPU time, equivalent to the sum of the *tms\_utime* and *tms\_cutime* fields returned by  
35151 the *times()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001 for the  
35152 process in which *utility* is executed.
- 35153 • The System CPU time, equivalent to the sum of the *tms\_stime* and *tms\_cstime* fields returned  
35154 by the *times()* function for the process in which *utility* is executed.

35155 The precision of the timing shall be no less than the granularity defined for the size of the clock  
35156 tick unit on the system, but the results shall be reported in terms of standard time units (for  
35157 example, 0.02 seconds, 00:00:00.02, 1m33.75s, 365.21 seconds), not numbers of clock ticks.

35158 When *time* is used as part of a pipeline, the times reported are unspecified, except when it is the  
35159 sole command within a grouping command (see Section 2.9.4.1 (on page 52)) in that pipeline. For  
35160 example, the commands on the left are unspecified; those on the right report on utilities **a** and **c**,  
35161 respectively:

35162 time a | b | c { time a } | b | c  
35163 a | b | time c a | b | (time c)

## 35164 OPTIONS

35165 The *time* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
35166 12.2, Utility Syntax Guidelines.

35167 The following option shall be supported:

35168 -p Write the timing output to standard error in the format shown in the STDERR  
35169 section.

## 35170 OPERANDS

35171 The following operands shall be supported:

35172 *utility* The name of a utility that is to be invoked. If the *utility* operand names any of the  
35173 special built-in utilities in Section 2.14 (on page 64), the results are undefined.

35174 *argument* Any string to be supplied as an argument when invoking the utility named by the  
35175 *utility* operand.

## 35176 STDIN

35177 Not used.

## 35178 INPUT FILES

35179 None.

**35180 ENVIRONMENT VARIABLES**

35181 The following environment variables shall affect the execution of *time*:

- 35182 **LANG** Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)
- 35186 **LC\_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.
- 35188 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
- 35191 **LC\_MESSAGES**  
Determine the locale that should be used to affect the format and contents of diagnostic and informative messages written to standard error.
- 35194 **LC\_NUMERIC**  
Determine the locale for numeric formatting.
- 35196 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 35197 **PATH** Determine the search path that shall be used to locate the utility to be invoked; see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

**35200 ASYNCHRONOUS EVENTS**

35201 Default.

**35202 STDOUT**

35203 Not used.

**35204 STDERR**

35205 The standard error shall be used to write the timing statistics. If **-p** is specified, the following  
35206 format shall be used in the POSIX locale:

35207 "real %f\nuser %f\nsys %f\n", <real seconds>, <user seconds>,  
35208 <system seconds>

35209 where each floating-point number shall be expressed in seconds. The precision used may be less  
35210 than the default six digits of %f, but shall be sufficiently precise to accommodate the size of the  
35211 clock tick on the system (for example, if there were 60 clock ticks per second, at least two digits  
35212 shall follow the radix character). The number of digits following the radix character shall be no  
35213 less than one, even if this always results in a trailing zero. The implementation may append  
35214 white space and additional information following the format shown here.

**35215 OUTPUT FILES**

35216 None.

**35217 EXTENDED DESCRIPTION**

35218 None.

**35219 EXIT STATUS**

35220 If the *utility* utility is invoked, the exit status of *time* shall be the exit status of *utility*; otherwise,  
35221 the *time* utility shall exit with one of the following values:

35222 1-125 An error occurred in the *time* utility.

- 35223        126    The utility specified by *utility* was found but could not be invoked.  
35224        127    The utility specified by *utility* could not be found.

## 35225 CONSEQUENCES OF ERRORS

35226        Default.

## 35227 APPLICATION USAGE

35228        The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if  
35229        an error occurs so that applications can distinguish “failure to find a utility” from “invoked  
35230        utility exited with an error indication”. The value 127 was chosen because it is not commonly  
35231        used for other meanings; most utilities use small values for “normal error conditions” and the  
35232        values above 128 can be confused with termination due to receipt of a signal. The value 126 was  
35233        chosen in a similar manner to indicate that the utility could be found, but not invoked. Some  
35234        scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
35235        between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
35236        *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
35237        any other reason.

## 35238 EXAMPLES

35239        It is frequently desirable to apply *time* to pipelines or lists of commands. This can be done by  
35240        placing pipelines and command lists in a single file; this file can then be invoked as a utility, and  
35241        the *time* applies to everything in the file.

35242        Alternatively, the following command can be used to apply *time* to a complex command:

35243        `time sh -c 'complex-command-line'`

## 35244 RATIONALE

35245        When the *time* utility was originally proposed to be included in the ISO POSIX-2:1993 standard,  
35246        questions were raised about its suitability for inclusion on the grounds that it was not useful for  
35247        conforming applications, specifically:

- The underlying CPU definitions from the System Interfaces volume of IEEE Std 1003.1-2001  
35249        are vague, so the numeric output could not be compared accurately between systems or even  
35250        between invocations.
- The creation of portable benchmark programs was outside the scope this volume of  
35252        IEEE Std 1003.1-2001.

35253        However, *time* does fit in the scope of user portability. Human judgement can be applied to the  
35254        analysis of the output, and it could be very useful in hands-on debugging of applications or in  
35255        providing subjective measures of system performance. Hence it has been included in this  
35256        volume of IEEE Std 1003.1-2001.

35257        The default output format has been left unspecified because historical implementations differ  
35258        greatly in their style of depicting this numeric output. The **-p** option was invented to provide  
35259        scripts with a common means of obtaining this information.

35260        In the KornShell, *time* is a shell reserved word that can be used to time an entire pipeline, rather  
35261        than just a simple command. The POSIX definition has been worded to allow this  
35262        implementation. Consideration was given to invalidating this approach because of the historical  
35263        model from the C shell and System V shell. However, since the System V *time* utility historically  
35264        has not produced accurate results in pipeline timing (because the constituent processes are not  
35265        all owned by the same parent process, as allowed by POSIX), it did not seem worthwhile to  
35266        break historical KornShell usage.

35267        The term *utility* is used, rather than *command*, to highlight the fact that shell compound  
35268        commands, pipelines, special built-ins, and so on, cannot be used directly. However, *utility*

35269 includes user application programs and shell scripts, not just the standard utilities.

**35270 FUTURE DIRECTIONS**

35271 None.

**35272 SEE ALSO**

35273 Chapter 2 (on page 29), *sh*, the System Interfaces volume of IEEE Std 1003.1-2001, *times()*

**35274 CHANGE HISTORY**

35275 First released in Issue 2.

**35276 Issue 6**

35277 This utility is marked as part of the User Portability Utilities option.

## 35278 NAME

35279 touch — change file access and modification times

## 35280 SYNOPSIS

35281 touch [-acm][ -r ref\_file| -t time] file...

## 35282 DESCRIPTION

35283 The *touch* utility shall change the modification times, access times, or both of files. The  
35284 modification time shall be equivalent to the value of the *st\_mtime* member of the **stat** structure  
35285 for a file, as described in the System Interfaces volume of IEEE Std 1003.1-2001; the access time  
35286 shall be equivalent to the value of *st\_atime*.

35287 The time used can be specified by the **-t time** option-argument, the corresponding time fields of  
35288 the file referenced by the **-r ref\_file** option-argument, or the *date\_time* operand, as specified in the  
35289 following sections. If none of these are specified, *touch* shall use the current time (the value  
35290 returned by the equivalent of the *time()* function defined in the System Interfaces volume of  
35291 IEEE Std 1003.1-2001).

35292 For each *file* operand, *touch* shall perform actions equivalent to the following functions defined  
35293 in the System Interfaces volume of IEEE Std 1003.1-2001:

- 35294 1. If *file* does not exist, a *creat()* function call is made with the *file* operand used as the *path*  
35295 argument and the value of the bitwise-inclusive OR of S\_IRUSR, S\_IWUSR, S\_IRGRP,  
35296 S\_IWGRP, S\_IROTH, and S\_IWOTH used as the *mode* argument.
- 35297 2. The *utime()* function is called with the following arguments:
  - 35298 a. The *file* operand is used as the *path* argument.
  - 35299 b. The **utimbuf** structure members *actime* and *modtime* are determined as described in  
35300 the OPTIONS section.

## 35301 OPTIONS

35302 The *touch* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
35303 12.2, Utility Syntax Guidelines.

35304 The following options shall be supported:

- 35305 **-a** Change the access time of *file*. Do not change the modification time unless **-m** is  
35306 also specified.
- 35307 **-c** Do not create a specified *file* if it does not exist. Do not write any diagnostic  
35308 messages concerning this condition.
- 35309 **-m** Change the modification time of *file*. Do not change the access time unless **-a** is  
35310 also specified.
- 35311 **-r ref\_file** Use the corresponding time of the file named by the pathname *ref\_file* instead of  
35312 the current time.
- 35313 **-t time** Use the specified *time* instead of the current time. The option-argument shall be a  
35314 decimal number of the form:  
  
35315  $[[CC]YY]MMDDhhmm[.SS]$   
35316 where each two digits represents the following:
  - 35317 **MM** The month of the year [01,12].
  - 35318 **DD** The day of the month [01,31].

35319            *hh*        The hour of the day [00,23].  
 35320            *mm*        The minute of the hour [00,59].  
 35321            *CC*        The first two digits of the year (the century).  
 35322            *YY*        The second two digits of the year.  
 35323            *SS*        The second of the minute [00,60].

35324            Both *CC* and *YY* shall be optional. If neither is given, the current year shall be assumed. If *YY* is specified, but *CC* is not, *CC* shall be derived as follows:

| If YY is: | <i>CC</i> becomes: |
|-----------|--------------------|
| [69,99]   | 19                 |
| [00,68]   | 20                 |

35329            **Note:** It is expected that in a future version of IEEE Std 1003.1-2001 the default century inferred from a 2-digit year will change. (This would apply to all commands accepting a 2-digit year as input.)

35332            The resulting time shall be affected by the value of the *TZ* environment variable. If  
 35333            the resulting time value precedes the Epoch, *touch* shall exit immediately with an  
 35334            error status. The range of valid times past the Epoch is implementation-defined,  
 35335            but it shall extend to at least the time 0 hours, 0 minutes, 0 seconds, January 1,  
 35336            2038, Coordinated Universal Time. Some implementations may not be able to  
 35337            represent dates beyond January 18, 2038, because they use **signed int** as a time  
 35338            holder.

35339            The range for *SS* is [00,60] rather than [00,59] because of leap seconds. If *SS* is 60,  
 35340            and the resulting time, as affected by the *TZ* environment variable, does not refer  
 35341            to a leap second, the resulting time shall be one second after a time where *SS* is 59.  
 35342            If *SS* is not given a value, it is assumed to be zero.

35343            If neither the **-a** nor **-m** options were specified, *touch* shall behave as if both the **-a** and **-m**  
 35344            options were specified.

## 35345 OPERANDS

35346            The following operands shall be supported:

35347            *file*        A pathname of a file whose times shall be modified.

## 35348 STDIN

35349            Not used.

## 35350 INPUT FILES

35351            None.

## 35352 ENVIRONMENT VARIABLES

35353            The following environment variables shall affect the execution of *touch*:

35354            *LANG*        Provide a default value for the internationalization variables that are unset or null.  
 35355            (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
 35356            Internationalization Variables for the precedence of internationalization variables  
 35357            used to determine the values of locale categories.)

35358            *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 35359            internationalization variables.

35360            *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
 35361            characters (for example, single-byte as opposed to multi-byte characters in

35362 arguments).

35363 ***LC\_MESSAGES***

35364 Determine the locale that should be used to affect the format and contents of

35365 diagnostic messages written to standard error.

35366 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

35367 ***TZ*** Determine the timezone to be used for interpreting the *time* option-argument. If *TZ*

35368 is unset or null, an unspecified default timezone shall be used.

35369 **ASYNCHRONOUS EVENTS**

35370 Default.

35371 **STDOUT**

35372 Not used.

35373 **STDERR**

35374 The standard error shall be used only for diagnostic messages.

35375 **OUTPUT FILES**

35376 None.

35377 **EXTENDED DESCRIPTION**

35378 None.

35379 **EXIT STATUS**

35380 The following exit values shall be returned:

35381 0 The utility executed successfully and all requested changes were made.

35382 >0 An error occurred.

35383 **CONSEQUENCES OF ERRORS**

35384 Default.

35385 **APPLICATION USAGE**

35386 The interpretation of time is taken to be *seconds since the Epoch* (see the Base Definitions volume

35387 of IEEE Std 1003.1-2001, Section 4.14, Seconds Since the Epoch). It should be noted that

35388 implementations conforming to the System Interfaces volume of IEEE Std 1003.1-2001 do not

35389 take leap seconds into account when computing seconds since the Epoch. When *SS*=60 is used,

35390 the resulting time always refers to 1 plus *seconds since the Epoch* for a time when *SS*=59.

35391 Although the **-t *time*** option-argument specifies values in 1969, the access time and modification

35392 time fields are defined in terms of seconds since the Epoch (00:00:00 on 1 January 1970 UTC).

35393 Therefore, depending on the value of *TZ* when *touch* is run, there is never more than a few valid

35394 hours in 1969 and there need not be any valid times in 1969.

35395 One ambiguous situation occurs if **-t *time*** is not specified, **-r *ref\_file*** is not specified, and the first

35396 operand is an eight or ten-digit decimal number. A portable script can avoid this problem by

35397 using:

35398 `touch -- file`

35399 or:

35400 `touch ./file`

35401 in this case.

**35402 EXAMPLES**

35403 None.

**35404 RATIONALE**

35405 The functionality of *touch* is described almost entirely through references to functions in the  
35406 System Interfaces volume of IEEE Std 1003.1-2001. In this way, there is no duplication of effort  
35407 required for describing such side effects as the relationship of user IDs to the user database,  
35408 permissions, and so on.

35409 There are some significant differences between the *touch* utility in this volume of  
35410 IEEE Std 1003.1-2001 and those in System V and BSD systems. They are upwards-compatible for  
35411 historical applications from both implementations:

- 35412 1. In System V, an ambiguity exists when a pathname that is a decimal number leads the  
35413 operands; it is treated as a time value. In BSD, no *time* value is allowed; files may only be  
35414 *touched* to the current time. The **-t time** construct solves these problems for future  
35415 conforming applications (note that the **-t** option is not historical practice).
- 35416 2. The inclusion of the century digits, *CC*, is also new. Note that a ten-digit *time* value is  
35417 treated as if *YY*, and not *CC*, were specified. The caveat about the range of dates following  
35418 the Epoch was included as recognition that some implementations are not able to  
35419 represent dates beyond 18 January 2038 because they use **signed int** as a time holder.

35420 The **-r** option was added because several comments requested this capability. This option was  
35421 named **-f** in an early proposal, but was changed because the **-f** option is used in the BSD version  
35422 of *touch* with a different meaning.

35423 At least one historical implementation of *touch* incremented the exit code if **-c** was specified and  
35424 the file did not exist. This volume of IEEE Std 1003.1-2001 requires exit status zero if no errors  
35425 occur.

**35426 FUTURE DIRECTIONS**

35427 Applications should use the **-r** or **-t** options.

**35428 SEE ALSO**

35429 *date*, the System Interfaces volume of IEEE Std 1003.1-2001, *creat()*, *time()*, *utime()*, the Base  
35430 Definitions volume of IEEE Std 1003.1-2001, **<sys/stat.h>**

**35431 CHANGE HISTORY**

35432 First released in Issue 2.

**35433 Issue 6**

35434 The obsolescent *date\_time* operand is removed.

35435 The Open Group Corrigendum U027/1 is applied. This extends the range of valid time past the  
35436 Epoch to at least the time 0 hours, 0 minutes, 0 seconds, January 1, 2038, Coordinated Universal  
35437 Time. This is a new requirement on POSIX implementations.

35438 The range for seconds is changed from [00,61] to [00,60] to align with the ISO/IEC 9899:1999  
35439 standard, and to allow for positive leap seconds.

**35440 NAME**

35441        *tput* — change terminal characteristics

**35442 SYNOPSIS**

35443 UP      *tput [-T type] operand...*

35444

**35445 DESCRIPTION**

35446        The *tput* utility shall display terminal-dependent information. The manner in which this  
35447        information is retrieved is unspecified. The information displayed shall clear the terminal screen,  
35448        initialize the user's terminal, or reset the user's terminal, depending on the operand given. The  
35449        exact consequences of displaying this information are unspecified.

**35450 OPTIONS**

35451        The *tput* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
35452        12.2, Utility Syntax Guidelines.

35453        The following option shall be supported:

35454        **-T type**      Indicate the type of terminal. If this option is not supplied and the *TERM* variable  
35455        is unset or null, an unspecified default terminal type shall be used. The setting of  
35456        *type* shall take precedence over the value in *TERM*.

**35457 OPERANDS**

35458        The following strings shall be supported as operands by the implementation in the POSIX locale:

35459        **clear**        Display the clear-screen sequence.

35460        **init**        Display the sequence that initializes the user's terminal in an implementation-  
35461        defined manner.

35462        **reset**        Display the sequence that resets the user's terminal in an implementation-defined  
35463        manner.

35464        If a terminal does not support any of the operations described by these operands, this shall not  
35465        be considered an error condition.

**35466 STDIN**

35467        Not used.

**35468 INPUT FILES**

35469        None.

**35470 ENVIRONMENT VARIABLES**

35471        The following environment variables shall affect the execution of *tput*:

35472        **LANG**        Provide a default value for the internationalization variables that are unset or null.  
35473        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
35474        Internationalization Variables for the precedence of internationalization variables  
35475        used to determine the values of locale categories.)

35476        **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
35477        internationalization variables.

35478        **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
35479        characters (for example, single-byte as opposed to multi-byte characters in  
35480        arguments).

**35481 *LC\_MESSAGES***

35482        Determine the locale that should be used to affect the format and contents of  
35483        diagnostic messages written to standard error.

|           |                               |                                                                                                                                                                   |
|-----------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 35484 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                             |
| 35485     | <b>TERM</b>                   | Determine the terminal type. If this variable is unset or null, and if the <b>-T</b> option is not specified, an unspecified default terminal type shall be used. |
| 35486     |                               |                                                                                                                                                                   |
| 35487     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                   |
| 35488     |                               | Default.                                                                                                                                                          |
| 35489     | <b>STDOUT</b>                 |                                                                                                                                                                   |
| 35490     |                               | If standard output is a terminal device, it may be used for writing the appropriate sequence to                                                                   |
| 35491     |                               | clear the screen or reset or initialize the terminal. If standard output is not a terminal device,                                                                |
| 35492     |                               | undefined results occur.                                                                                                                                          |
| 35493     | <b>STDERR</b>                 |                                                                                                                                                                   |
| 35494     |                               | The standard error shall be used only for diagnostic messages.                                                                                                    |
| 35495     | <b>OUTPUT FILES</b>           |                                                                                                                                                                   |
| 35496     |                               | None.                                                                                                                                                             |
| 35497     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                   |
| 35498     |                               | None.                                                                                                                                                             |
| 35499     | <b>EXIT STATUS</b>            |                                                                                                                                                                   |
| 35500     |                               | The following exit values shall be returned:                                                                                                                      |
| 35501     | 0                             | The requested string was written successfully.                                                                                                                    |
| 35502     | 1                             | Unspecified.                                                                                                                                                      |
| 35503     | 2                             | Usage error.                                                                                                                                                      |
| 35504     | 3                             | No information is available about the specified terminal type.                                                                                                    |
| 35505     | 4                             | The specified operand is invalid.                                                                                                                                 |
| 35506     | >4                            | An error occurred.                                                                                                                                                |
| 35507     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                   |
| 35508     |                               | If one of the operands is not available for the terminal, <i>tput</i> continues processing the remaining                                                          |
| 35509     |                               | operands.                                                                                                                                                         |
| 35510     | <b>APPLICATION USAGE</b>      |                                                                                                                                                                   |
| 35511     |                               | The difference between resetting and initializing a terminal is left unspecified, as they vary                                                                    |
| 35512     |                               | greatly based on hardware types. In general, resetting is a more severe action.                                                                                   |
| 35513     |                               | Some terminals use control characters to perform the stated functions, and on such terminals it                                                                   |
| 35514     |                               | might make sense to use <i>tput</i> to store the initialization strings in a file or environment variable                                                         |
| 35515     |                               | for later use. However, because other terminals might rely on system calls to do this work, the                                                                   |
| 35516     |                               | standard output cannot be used in a portable manner, such as the following non-portable                                                                           |
| 35517     |                               | constructs:                                                                                                                                                       |
| 35518     |                               | ClearVar='tput clear'                                                                                                                                             |
| 35519     |                               | tput reset   mailx -s "Wake Up" ddg                                                                                                                               |
| 35520     | <b>EXAMPLES</b>               |                                                                                                                                                                   |
| 35521     | 1.                            | Initialize the terminal according to the type of terminal in the environmental variable                                                                           |
| 35522     |                               | <b>TERM</b> . This command can be included in a <b>.profile</b> file.                                                                                             |
| 35523     |                               | tput init                                                                                                                                                         |
| 35524     | 2.                            | Reset a 450 terminal.                                                                                                                                             |

35525                    tput -T 450 reset

## 35526 RATIONALE

35527        The list of operands was reduced to a minimum for the following reasons:

- 35528        • The only features chosen were those that were likely to be used by human users interacting with a terminal.
- 35529        • Specifying the full *terminfo* set was not considered desirable, but the standard developers did not want to select among operands.
- 35530        • This volume of IEEE Std 1003.1-2001 does not attempt to provide applications with sophisticated terminal handling capabilities, as that falls outside of its assigned scope and intersects with the responsibilities of other standards bodies.

35535        The difference between resetting and initializing a terminal is left unspecified as this varies greatly based on hardware types. In general, resetting is a more severe action.

35537        The exit status of 1 is historically reserved for finding out if a Boolean operand is not set. Although the operands were reduced to a minimum, the exit status of 1 should still be reserved for the Boolean operands, for those sites that wish to support them.

## 35540 FUTURE DIRECTIONS

35541        None.

## 35542 SEE ALSO

35543        *stty*, *tabs*

## 35544 CHANGE HISTORY

35545        First released in Issue 4.

## 35546 Issue 6

35547        This utility is marked as part of the User Portability Utilities option.

**35548 NAME**

35549 tr — translate characters

**35550 SYNOPSIS**

```
35551 tr [-c | -C][-s] string1 string2
35552 tr -s [-c | -C] string1
35553 tr -d [-c | -C] string1
35554 tr -ds [-c | -C] string1 string2
```

**35555 DESCRIPTION**

35556 The *tr* utility shall copy the standard input to the standard output with substitution or deletion of selected characters. The options specified and the *string1* and *string2* operands shall control translations that occur while copying characters and single-character collating elements.

**35559 OPTIONS**

35560 The *tr* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

35562 The following options shall be supported:

- |                 |                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------|
| 35563 <b>-c</b> | Complement the set of values specified by <i>string1</i> . See the EXTENDED DESCRIPTION section.                    |
| 35564           |                                                                                                                     |
| 35565 <b>-C</b> | Complement the set of characters specified by <i>string1</i> . See the EXTENDED DESCRIPTION section.                |
| 35566           |                                                                                                                     |
| 35567 <b>-d</b> | Delete all occurrences of input characters that are specified by <i>string1</i> .                                   |
| 35568 <b>-s</b> | Replace instances of repeated characters with a single character, as described in the EXTENDED DESCRIPTION section. |
| 35569           |                                                                                                                     |

**35570 OPERANDS**

35571 The following operands shall be supported:

35572   *string1, string2*

35573         Translation control strings. Each string shall represent a set of characters to be converted into an array of characters used for the translation. For a detailed description of how the strings are interpreted, see the EXTENDED DESCRIPTION section.

**35577 STDIN**

35578 The standard input can be any type of file.

**35579 INPUT FILES**

35580 None.

**35581 ENVIRONMENT VARIABLES**

35582 The following environment variables shall affect the execution of *tr*:

- |                         |                                                                                                                                                                                                                                                                                                       |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 35583 <b>LANG</b>       | Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.) |
| 35584                   |                                                                                                                                                                                                                                                                                                       |
| 35585                   |                                                                                                                                                                                                                                                                                                       |
| 35586                   |                                                                                                                                                                                                                                                                                                       |
| 35587 <b>LC_ALL</b>     | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                              |
| 35588                   |                                                                                                                                                                                                                                                                                                       |
| 35589 <b>LC_COLLATE</b> | Determine the locale for the behavior of range expressions and equivalence classes.                                                                                                                                                                                                                   |
| 35590                   |                                                                                                                                                                                                                                                                                                       |

|           |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 35591     | <i>LC_CTYPE</i>             | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments) and the behavior of character classes.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35592     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35593     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35594     | <i>LC_MESSAGES</i>          | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 35595     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35596     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35597 XSI | <i>NLSPATH</i>              | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 35598     | <b>ASYNCHRONOUS EVENTS</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35599     |                             | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 35600     | <b>STDOUT</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35601     |                             | The <i>tr</i> output shall be identical to the input, with the exception of the specified transformations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 35602     | <b>STDERR</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35603     |                             | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 35604     | <b>OUTPUT FILES</b>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35605     |                             | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 35606     | <b>EXTENDED DESCRIPTION</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35607     |                             | The operands <i>string1</i> and <i>string2</i> (if specified) define two arrays of characters. The constructs in                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 35608     |                             | the following list can be used to specify characters or single-character collating elements. If any                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 35609     |                             | of the constructs result in multi-character collating elements, <i>tr</i> shall exclude, without a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35610     |                             | diagnostic, those multi-character elements from the resulting array.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 35611     | <i>character</i>            | Any character not described by one of the conventions below shall represent itself.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 35612     | <i>\octal</i>               | Octal sequences can be used to represent characters with specific coded values. An octal sequence shall consist of a backslash followed by the longest sequence of one, two, or three-octal-digit characters (01234567). The sequence shall cause the value whose encoding is represented by the one, two, or three-digit octal integer to be placed into the array. If the size of a byte on the system is greater than nine bits, the valid escape sequence used to represent a byte is implementation-defined. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading '\ ' for each byte. |
| 35613     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35614     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35615     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35616     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35617     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35618     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35619     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35620     | <i>\character</i>           | The backslash-escape sequences 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') shall be supported. The results of using any other character, other than an octal digit, following the backslash are unspecified.                                                                                                                                                                                                                                                                                                             |
| 35621     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35622     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35623     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35624     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35625     | <i>c–c</i>                  | In the POSIX locale, this construct shall represent the range of collating elements between the range endpoints (as long as neither endpoint is an octal sequence of the form <i>\octal</i> ), inclusive, as defined by the collation sequence. The characters or collating elements in the range shall be placed in the array in ascending collation sequence. If the second endpoint precedes the starting endpoint in the collation sequence, it is unspecified whether the range of collating elements is empty, or this construct is treated as invalid. In locales other than the POSIX locale, this construct has unspecified behavior.  |
| 35626     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35627     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35628     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35629     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35630     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35631     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35632     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 35633     |                             | If either or both of the range endpoints are octal sequences of the form <i>\octal</i> , this                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35634     |                             | shall represent the range of specific coded values between the two range                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 35635     |                             | endpoints, inclusive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

35636           [:class:]     Represents all characters belonging to the defined character class, as defined by the  
 35637           current setting of the *LC\_CTYPE* locale category. The following character class  
 35638           names shall be accepted when specified in *string1*:

35639           alnum   blank   digit   lower   punct   upper  
 35640           alpha   cntrl   graph   print   space   xdigit

35641 XSI       In addition, character class expressions of the form [:name:] shall be recognized in  
 35642           those locales where the *name* keyword has been given a **charclass** definition in the  
 35643           *LC\_CTYPE* category.

35644           When both the **-d** and **-s** options are specified, any of the character class names  
 35645           shall be accepted in *string2*. Otherwise, only character class names **lower** or **upper**  
 35646           are valid in *string2* and then only if the corresponding character class (**upper** and  
 35647           **lower**, respectively) is specified in the same relative position in *string1*. Such a  
 35648           specification shall be interpreted as a request for case conversion. When [:lower:]  
 35649           appears in *string1* and [:upper:] appears in *string2*, the arrays shall contain the  
 35650           characters from the **toupper** mapping in the *LC\_CTYPE* category of the current  
 35651           locale. When [:upper:] appears in *string1* and [:lower:] appears in *string2*, the arrays  
 35652           shall contain the characters from the **tolower** mapping in the *LC\_CTYPE* category  
 35653           of the current locale. The first character from each mapping pair shall be in the  
 35654           array for *string1* and the second character from each mapping pair shall be in the  
 35655           array for *string2* in the same relative position.

35656           Except for case conversion, the characters specified by a character class expression  
 35657           shall be placed in the array in an unspecified order.

35658           If the name specified for *class* does not define a valid character class in the current  
 35659           locale, the behavior is undefined.

35660           [=equiv=]   Represents all characters or collating elements belonging to the same equivalence  
 35661           class as *equiv*, as defined by the current setting of the *LC\_COLLATE* locale  
 35662           category. An equivalence class expression shall be allowed only in *string1*, or in  
 35663           *string2* when it is being used by the combined **-d** and **-s** options. The characters  
 35664           belonging to the equivalence class shall be placed in the array in an unspecified  
 35665           order.

35666           [x\*n]     Represents *n* repeated occurrences of the character *x*. Because this expression is  
 35667           used to map multiple characters to one, it is only valid when it occurs in *string2*. If  
 35668           *n* is omitted or is zero, it shall be interpreted as large enough to extend the *string2*-  
 35669           based sequence to the length of the *string1*-based sequence. If *n* has a leading zero,  
 35670           it shall be interpreted as an octal value. Otherwise, it shall be interpreted as a  
 35671           decimal value.

35672           When the **-d** option is not specified:

- Each input character found in the array specified by *string1* shall be replaced by the character  
     in the same relative position in the array specified by *string2*. When the array specified by  
     *string2* is shorter than the one specified by *string1*, the results are unspecified.
- If the **-C** option is specified, the complements of the characters specified by *string1* (the set of  
     all characters in the current character set, as defined by the current setting of *LC\_CTYPE*,  
     except for those actually specified in the *string1* operand) shall be placed in the array in  
     ascending collation sequence, as defined by the current setting of *LC\_COLLATE*.
- If the **-c** option is specified, the complement of the values specified by *string1* shall be placed  
     in the array in ascending order by binary value.

- 35682     • Because the order in which characters specified by character class expressions or equivalence  
35683        class expressions is undefined, such expressions should only be used if the intent is to map  
35684        several characters into one. An exception is case conversion, as described previously.

35685 When the **-d** option is specified:

- 35686     • Input characters found in the array specified by *string1* shall be deleted.
- 35687     • When the **-C** option is specified with **-d**, all characters except those specified by *string1* shall  
35688        be deleted. The contents of *string2* are ignored, unless the **-s** option is also specified.
- 35689     • When the **-c** option is specified with **-d**, all values except those specified by *string1* shall be  
35690        deleted. The contents of *string2* shall be ignored, unless the **-s** option is also specified.
- 35691     • The same string cannot be used for both the **-d** and the **-s** option; when both options are  
35692        specified, both *string1* (used for deletion) and *string2* (used for squeezing) shall be required.

35693 When the **-s** option is specified, after any deletions or translations have taken place, repeated  
35694 sequences of the same character shall be replaced by one occurrence of the same character, if the  
35695 character is found in the array specified by the last operand. If the last operand contains a  
35696 character class, such as the following example:

35697 `tr -s '[ :space:]'`

35698 the last operand's array shall contain all of the characters in that character class. However, in a  
35699 case conversion, as described previously, such as:

35700 `tr -s '[ :upper:]' '[ :lower:]'`

35701 the last operand's array shall contain only those characters defined as the second characters in  
35702 each of the **toupper** or **tolower** character pairs, as appropriate.

35703 An empty string used for *string1* or *string2* produces undefined results.

#### 35704 EXIT STATUS

35705 The following exit values shall be returned:

35706     0 All input was processed successfully.

35707     >0 An error occurred.

#### 35708 CONSEQUENCES OF ERRORS

35709 Default.

#### 35710 APPLICATION USAGE

35711 If necessary, *string1* and *string2* can be quoted to avoid pattern matching by the shell.

35712 If an ordinary digit (representing itself) is to follow an octal sequence, the octal sequence must  
35713 use the full three digits to avoid ambiguity.

35714 When *string2* is shorter than *string1*, a difference results between historical System V and BSD  
35715 systems. A BSD system pads *string2* with the last character found in *string2*. Thus, it is possible  
35716 to do the following:

35717 `tr 0123456789 d`

35718 which would translate all digits to the letter '`d`'. Since this area is specifically unspecified in  
35719 this volume of IEEE Std 1003.1-2001, both the BSD and System V behaviors are allowed, but a  
35720 conforming application cannot rely on the BSD behavior. It would have to code the example in  
35721 the following way:

35722 `tr 0123456789 '[d*]'`

35723 It should be noted that, despite similarities in appearance, the string operands used by *tr* are not  
35724 regular expressions.

35725 Unlike some historical implementations, this definition of the *tr* utility correctly processes NUL  
35726 characters in its input stream. NUL characters can be stripped by using:

35727 `tr -d '\000'`

## 35728 EXAMPLES

- 35729 1. The following example creates a list of all words in **file1** one per line in **file2**, where a word  
35730 is taken to be a maximal string of letters.

35731 `tr -cs "[[:alpha:]]" "[\n*]" <file1 >file2`

- 35732 2. The next example translates all lowercase characters in **file1** to uppercase and writes the  
35733 results to standard output.

35734 `tr "[[:lower:]]" "[[:upper:]]" <file1`

- 35735 3. This example uses an equivalence class to identify accented variants of the base character  
35736 'e' in **file1**, which are stripped of diacritical marks and written to **file2**.

35737 `tr "[=e=]" e <file1 >file2`

## 35738 RATIONALE

35739 In some early proposals, an explicit option **-n** was added to disable the historical behavior of  
35740 stripping NUL characters from the input. It was considered that automatically stripping NUL  
35741 characters from the input was not correct functionality. However, the removal of **-n** in a later  
35742 proposal does not remove the requirement that *tr* correctly process NUL characters in its input  
35743 stream. NUL characters can be stripped by using *tr -d '\000'*.

35744 Historical implementations of *tr* differ widely in syntax and behavior. For example, the BSD  
35745 version has not needed the bracket characters for the repetition sequence. The *tr* utility syntax is  
35746 based more closely on the System V and XPG3 model while attempting to accommodate  
35747 historical BSD implementations. In the case of the short *string2* padding, the decision was to  
35748 unspecify the behavior and preserve System V and XPG3 scripts, which might find difficulty  
35749 with the BSD method. The assumption was made that BSD users of *tr* have to make  
35750 accommodations to meet the syntax defined here. Since it is possible to use the repetition  
35751 sequence to duplicate the desired behavior, whereas there is no simple way to achieve the  
35752 System V method, this was the correct, if not desirable, approach.

35753 The use of octal values to specify control characters, while having historical precedents, is not  
35754 portable. The introduction of escape sequences for control characters should provide the  
35755 necessary portability. It is recognized that this may cause some historical scripts to break.

35756 An early proposal included support for multi-character collating elements. It was pointed out  
35757 that, while *tr* does employ some syntactical elements from REs, the aim of *tr* is quite different;  
35758 ranges, for example, do not have a similar meaning ("any of the chars in the range matches",  
35759 versus "translate each character in the range to the output counterpart"). As a result, the  
35760 previously included support for multi-character collating elements has been removed. What  
35761 remains are ranges in current collation order (to support, for example, accented characters),  
35762 character classes, and equivalence classes.

35763 In XPG3 the `[:class:]` and `[=equiv=]` conventions are shown with double brackets, as in RE syntax.  
35764 However, *tr* does not implement RE principles; it just borrows part of the syntax. Consequently,  
35765 `[:class:]` and `[=equiv=]` should be regarded as syntactical elements on a par with `[x*n]`, which is  
35766 not an RE bracket expression.

35767 The standard developers will consider changes to *tr* that allow it to translate characters between  
35768 different character encodings, or they will consider providing a new utility to accomplish this.

35769 On historical System V systems, a range expression requires enclosing square-brackets, such as:

35770 `tr '[a-z]' '[A-Z]'`

35771 However, BSD-based systems did not require the brackets, and this convention is used here to  
35772 avoid breaking large numbers of BSD scripts:

35773 `tr a-z A-Z`

35774 The preceding System V script will continue to work because the brackets, treated as regular  
35775 characters, are translated to themselves. However, any System V script that relied on "a-z"  
35776 representing the three characters 'a', 'z', and 'z' have to be rewritten as "az-".

35777 The ISO POSIX-2:1993 standard had a `-c` option that behaved similarly to the `-C` option, but did  
35778 not supply functionality equivalent to the `-c` option specified in IEEE Std 1003.1-2001. This  
35779 meant that historical practice of being able to specify `tr -d\200-\377` (which would delete all  
35780 bytes with the top bit set) would have no effect because, in the C locale, bytes with the values  
35781 octal 200 to octal 377 are not characters.

35782 The earlier version also said that octal sequences referred to collating elements and could be  
35783 placed adjacent to each other to specify multi-byte characters. However, it was noted that this  
35784 caused ambiguities because *tr* would not be able to tell whether adjacent octal sequences were  
35785 intending to specify multi-byte characters or multiple single byte characters.  
35786 IEEE Std 1003.1-2001 specifies that octal sequences always refer to single byte binary values.

## 35787 FUTURE DIRECTIONS

35788 None.

## 35789 SEE ALSO

35790 `sed`

## 35791 CHANGE HISTORY

35792 First released in Issue 2.

## 35793 Issue 6

35794 The `-C` operand is added, and the description of the `-c` operand is changed to align with the  
35795 IEEE P1003.2b draft standard.

35796 The normative text is reworded to avoid use of the term "must" for application requirements.

**35797 NAME**

35798 true — return true value

**35799 SYNOPSIS**

35800 true

**35801 DESCRIPTION**

35802 The *true* utility shall return with exit code zero.

**35803 OPTIONS**

35804 None.

**35805 OPERANDS**

35806 None.

**35807 STDIN**

35808 Not used.

**35809 INPUT FILES**

35810 None.

**35811 ENVIRONMENT VARIABLES**

35812 None.

**35813 ASYNCHRONOUS EVENTS**

35814 Default.

**35815 STDOUT**

35816 Not used.

**35817 STDERR**

35818 None.

**35819 OUTPUT FILES**

35820 None.

**35821 EXTENDED DESCRIPTION**

35822 None.

**35823 EXIT STATUS**

35824 Default.

**35825 CONSEQUENCES OF ERRORS**

35826 None.

**35827 APPLICATION USAGE**

35828 This utility is typically used in shell scripts, as shown in the EXAMPLES section. The special  
35829 built-in utility : is sometimes more efficient than *true*.

**35830 EXAMPLES**

35831 This command is executed forever:

35832 while true

35833 do

35834       command

35835 done

**35836 RATIONALE**

35837        The *true* utility has been retained in this volume of IEEE Std 1003.1-2001, even though the shell  
35838        special built-in : provides similar functionality, because *true* is widely used in historical scripts  
35839        and is less cryptic to novice script readers.

**35840 FUTURE DIRECTIONS**

35841        None.

**35842 SEE ALSO**

35843        *false*, Section 2.9 (on page 47)

**35844 CHANGE HISTORY**

35845        First released in Issue 2.

**35846 NAME**

35847        tsort — topological sort

**35848 SYNOPSIS**

35849 XSI     tsort [file]

35850

**35851 DESCRIPTION**

35852        The *tsort* utility shall write to standard output a totally ordered list of items consistent with a  
35853        partial ordering of items contained in the input.

35854        The application shall ensure that the input consists of pairs of items (non-empty strings)  
35855        separated by <blank>s. Pairs of different items indicate ordering. Pairs of identical items  
35856        indicate presence, but not ordering.

**35857 OPTIONS**

35858        None.

**35859 OPERANDS**

35860        The following operand shall be supported:

35861        *file*        A pathname of a text file to order. If no *file* operand is given, the standard input  
35862        shall be used.

**35863 STDIN**

35864        The standard input shall be a text file that is used if no *file* operand is given.

**35865 INPUT FILES**

35866        The input file named by the *file* operand is a text file.

**35867 ENVIRONMENT VARIABLES**

35868        The following environment variables shall affect the execution of *tsort*:

35869        *LANG*      Provide a default value for the internationalization variables that are unset or null.  
35870                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
35871                  Internationalization Variables for the precedence of internationalization variables  
35872                  used to determine the values of locale categories.)

35873        *LC\_ALL*    If set to a non-empty string value, override the values of all the other  
35874                  internationalization variables.

35875        *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
35876                  characters (for example, single-byte as opposed to multi-byte characters in  
35877                  arguments and input files).

**35878 *LC\_MESSAGES***

35879                  Determine the locale that should be used to affect the format and contents of  
35880                  diagnostic messages written to standard error.

35881        *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**35882 ASYNCHRONOUS EVENTS**

35883        Default.

**35884 STDOUT**

35885        The standard output shall be a text file consisting of the order list produced from the partially  
35886        ordered input.

**35887 STDERR**

35888 The standard error shall be used only for diagnostic messages.

**35889 OUTPUT FILES**

35890 None.

**35891 EXTENDED DESCRIPTION**

35892 None.

**35893 EXIT STATUS**

35894 The following exit values shall be returned:

35895 0 Successful completion.

35896 >0 An error occurred.

**35897 CONSEQUENCES OF ERRORS**

35898 Default.

**35899 APPLICATION USAGE**

35900 The *LC\_COLLATE* variable need not affect the actions of *tsort*. The output ordering is not lexicographic, but depends on the pairs of items given as input.

**35902 EXAMPLES**

35903 The command:

```
35904 tsort <<EOF
35905 a b c c d e
35906 g g
35907 f g e f
35908 h h
35909 EOF
```

35910 produces the output:

```
35911 a
35912 b
35913 c
35914 d
35915 e
35916 f
35917 g
35918 h
```

**35919 RATIONALE**

35920 None.

**35921 FUTURE DIRECTIONS**

35922 None.

**35923 SEE ALSO**

35924 None.

**35925 CHANGE HISTORY**

35926 First released in Issue 2.

**35927 Issue 6**

35928 The normative text is reworded to avoid use of the term “must” for application requirements.

**35929 NAME**

35930        *tty* — return user's terminal name

**35931 SYNOPSIS**

35932        *tty*

**35933 DESCRIPTION**

35934        The *tty* utility shall write to the standard output the name of the terminal that is open as  
35935        standard input. The name that is used shall be equivalent to the string that would be returned by  
35936        the *ttynname()* function defined in the System Interfaces volume of IEEE Std 1003.1-2001.

**35937 OPTIONS**

35938        The *tty* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
35939        Utility Syntax Guidelines.

**35940 OPERANDS**

35941        None.

**35942 STDIN**

35943        While no input is read from standard input, standard input shall be examined to determine  
35944        whether or not it is a terminal, and, if so, to determine the name of the terminal.

**35945 INPUT FILES**

35946        None.

**35947 ENVIRONMENT VARIABLES**

35948        The following environment variables shall affect the execution of *tty*:

35949        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
35950                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
35951                  Internationalization Variables for the precedence of internationalization variables  
35952                  used to determine the values of locale categories.)

35953        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
35954                  internationalization variables.

35955        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
35956                  characters (for example, single-byte as opposed to multi-byte characters in  
35957                  arguments).

**35958        *LC\_MESSAGES***

35959                  Determine the locale that should be used to affect the format and contents of  
35960                  diagnostic messages written to standard error and informative messages written to  
35961                  standard output.

35962 XSI        *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**35963 ASYNCHRONOUS EVENTS**

35964        Default.

**35965 STDOUT**

35966        If standard input is a terminal device, a pathname of the terminal as specified by the *ttynname()*  
35967                  function defined in the System Interfaces volume of IEEE Std 1003.1-2001 shall be written in the  
35968                  following format:

35969        "%s\n", <terminal name>

35970        Otherwise, a message shall be written indicating that standard input is not connected to a  
35971                  terminal. In the POSIX locale, the *tty* utility shall use the format:

35972 "not a tty\n"

### 35973 **STDERR**

35974 The standard error shall be used only for diagnostic messages.

### 35975 **OUTPUT FILES**

35976 None.

### 35977 **EXTENDED DESCRIPTION**

35978 None.

### 35979 **EXIT STATUS**

35980 The following exit values shall be returned:

35981 0 Standard input is a terminal.

35982 1 Standard input is not a terminal.

35983 >1 An error occurred.

### 35984 **CONSEQUENCES OF ERRORS**

35985 Default.

### 35986 **APPLICATION USAGE**

35987 This utility checks the status of the file open as standard input against that of an implementation-defined set of files. It is possible that no match can be found, or that the match found need not be the same file as that which was opened for standard input (although they are the same device).

### 35991 **EXAMPLES**

35992 None.

### 35993 **RATIONALE**

35994 None.

### 35995 **FUTURE DIRECTIONS**

35996 None.

### 35997 **SEE ALSO**

35998 The System Interfaces volume of IEEE Std 1003.1-2001, *isatty()*, *ttynname()*

### 35999 **CHANGE HISTORY**

36000 First released in Issue 2.

#### 36001 **Issue 5**

36002 The SYNOPSIS is changed to indicate two forms of the command, with the second form marked as obsolete. This is a clarification and does not change the functionality published in previous issues.

#### 36005 **Issue 6**

36006 The obsolescent **-s** option is removed.

36007 **NAME**

36008        type — write a description of command type

36009 **SYNOPSIS**

36010 XSI     type name...

36011

36012 **DESCRIPTION**

36013        The *type* utility shall indicate how each argument would be interpreted if used as a command name.

36015 **OPTIONS**

36016        None.

36017 **OPERANDS**

36018        The following operand shall be supported:

36019        *name*        A name to be interpreted.

36020 **STDIN**

36021        Not used.

36022 **INPUT FILES**

36023        None.

36024 **ENVIRONMENT VARIABLES**

36025        The following environment variables shall affect the execution of *type*:

36026        *LANG*        Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36030        *LC\_ALL*      If set to a non-empty string value, override the values of all the other internationalization variables.

36032        *LC\_CTYPE*    Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

36035        *LC\_MESSAGES*

36036        Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

36038        *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

36039        *PATH*        Determine the location of *name*, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

36041 **ASYNCHRONOUS EVENTS**

36042        Default.

36043 **STDOUT**

36044        The standard output of *type* contains information about each operand in an unspecified format. The information provided typically identifies the operand as a shell built-in, function, alias, or keyword, and where applicable, may display the operand's pathname.

**36047 STDERR**

36048 The standard error shall be used only for diagnostic messages.

**36049 OUTPUT FILES**

36050 None.

**36051 EXTENDED DESCRIPTION**

36052 None.

**36053 EXIT STATUS**

36054 The following exit values shall be returned:

36055 0 Successful completion.

36056 >0 An error occurred.

**36057 CONSEQUENCES OF ERRORS**

36058 Default.

**36059 APPLICATION USAGE**

36060 Since *type* must be aware of the contents of the current shell execution environment (such as the lists of commands, functions, and built-ins processed by *hash*), it is always provided as a shell regular built-in. If it is called in a separate utility execution environment, such as one of the following:

36064 nohup type writer

36065 find . -type f | xargs type

36066 it might not produce accurate results.

**36067 EXAMPLES**

36068 None.

**36069 RATIONALE**

36070 None.

**36071 FUTURE DIRECTIONS**

36072 None.

**36073 SEE ALSO**

36074 *command, hash*

**36075 CHANGE HISTORY**

36076 First released in Issue 2.

**36077 NAME**

36078        **ulimit** — set or report file size limit

**36079 SYNOPSIS**

36080 XSI     **ulimit [-f][blocks]**

36081

**36082 DESCRIPTION**

36083        The **ulimit** utility shall set or report the file-size writing limit imposed on files written by the shell and its child processes (files of any size may be read). Only a process with appropriate privileges can increase the limit.

**36086 OPTIONS**

36087        The **ulimit** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

36089        The following option shall be supported:

36090        **-f**        Set (or report, if no *blocks* operand is present), the file size limit in blocks. The **-f** option shall also be the default case.

**36092 OPERANDS**

36093        The following operand shall be supported:

36094        *blocks*      The number of 512-byte blocks to use as the new file size limit.

**36095 STDIN**

36096        Not used.

**36097 INPUT FILES**

36098        None.

**36099 ENVIRONMENT VARIABLES**

36100        The following environment variables shall affect the execution of **ulimit**:

36101        **LANG**      Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36105        **LC\_ALL**     If set to a non-empty string value, override the values of all the other internationalization variables.

36107        **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

**36110 *LC\_MESSAGES***

36111        Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

36113        **NLSPATH**    Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

**36114 ASYNCHRONOUS EVENTS**

36115        Default.

**36116 STDOUT**

36117        The standard output shall be used when no *blocks* operand is present. If the current number of blocks is limited, the number of blocks in the current limit shall be written in the following format:

36120        "%d\n", <number of 512-byte blocks>  
36121        If there is no current limit on the number of blocks, in the POSIX locale the following format  
36122        shall be used:  
36123        "unlimited\n"

**36124 STDERR**

36125        The standard error shall be used only for diagnostic messages.

**36126 OUTPUT FILES**

36127        None.

**36128 EXTENDED DESCRIPTION**

36129        None.

**36130 EXIT STATUS**

36131        The following exit values shall be returned:

36132        0      Successful completion.

36133        >0     A request for a higher limit was rejected or an error occurred.

**36134 CONSEQUENCES OF ERRORS**

36135        Default.

**36136 APPLICATION USAGE**

36137        Since *ulimit* affects the current shell execution environment, it is always provided as a shell  
36138        regular built-in. If it is called in a separate utility execution environment, such as one of the  
36139        following:

36140        nohup ulimit -f 10000

36141        env ulimit 10000

36142        it does not affect the file size limit of the caller's environment.

36143        Once a limit has been decreased by a process, it cannot be increased (unless appropriate  
36144        privileges are involved), even back to the original system limit.

**36145 EXAMPLES**

36146        Set the file size limit to 51 200 bytes:

36147        ulimit -f 100

**36148 RATIONALE**

36149        None.

**36150 FUTURE DIRECTIONS**

36151        None.

**36152 SEE ALSO**

36153        The System Interfaces volume of IEEE Std 1003.1-2001, *ulimit()*

**36154 CHANGE HISTORY**

36155        First released in Issue 2.

## 36156 NAME

36157        umask — get or set the file mode creation mask

## 36158 SYNOPSIS

36159        umask [-S][*mask*]

## 36160 DESCRIPTION

36161        The *umask* utility shall set the file mode creation mask of the current shell execution environment (see Section 2.12 (on page 61)) to the value specified by the *mask* operand. This mask shall affect the initial value of the file permission bits of subsequently created files. If *umask* is called in a subshell or separate utility execution environment, such as one of the following:

36165        (umask 002)  
36166        nohup umask ...  
36167        find . -exec umask ... \;

36168        it shall not affect the file mode creation mask of the caller's environment.

36169        If the *mask* operand is not specified, the *umask* utility shall write to standard output the value of  
36170        the invoking process' file mode creation mask.

## 36171 OPTIONS

36172        The *umask* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
36173        12.2, Utility Syntax Guidelines.

36174        The following option shall be supported:

36175        **-S**        Produce symbolic output.

36176        The default output style is unspecified, but shall be recognized on a subsequent invocation of  
36177        *umask* on the same system as a *mask* operand to restore the previous file mode creation mask.

## 36178 OPERANDS

36179        The following operand shall be supported:

36180        **mask**        A string specifying the new file mode creation mask. The string is treated in the  
36181        same way as the *mode* operand described in the EXTENDED DESCRIPTION  
36182        section for *chmod*.

36183        For a *symbolic\_mode* value, the new value of the file mode creation mask shall be  
36184        the logical complement of the file permission bits portion of the file mode specified  
36185        by the *symbolic\_mode* string.

36186        In a *symbolic\_mode* value, the permissions *op* characters '+' and '-' shall be  
36187        interpreted relative to the current file mode creation mask; '+' shall cause the bits  
36188        for the indicated permissions to be cleared in the mask; '-' shall cause the bits for  
36189        the indicated permissions to be set in the mask.

36190        The interpretation of *mode* values that specify file mode bits other than the file  
36191        permission bits is unspecified.

36192        In the octal integer form of *mode*, the specified bits are set in the file mode creation  
36193        mask.

36194        The file mode creation mask shall be set to the resulting numeric value.

36195        The default output of a prior invocation of *umask* on the same system with no  
36196        operand also shall be recognized as a *mask* operand.

**36197 STDIN**

36198 Not used.

**36199 INPUT FILES**

36200 None.

**36201 ENVIRONMENT VARIABLES**

36202 The following environment variables shall affect the execution of *umask*:

36203 **LANG** Provide a default value for the internationalization variables that are unset or null.  
36204 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
36205 Internationalization Variables for the precedence of internationalization variables  
36206 used to determine the values of locale categories.)

36207 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
36208 internationalization variables.

36209 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
36210 characters (for example, single-byte as opposed to multi-byte characters in  
36211 arguments).

**36212 LC\_MESSAGES**

36213 Determine the locale that should be used to affect the format and contents of  
36214 diagnostic messages written to standard error.

36215 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**36216 ASYNCHRONOUS EVENTS**

36217 Default.

**36218 STDOUT**

36219 When the *mask* operand is not specified, the *umask* utility shall write a message to standard  
36220 output that can later be used as a *umask mask* operand.

36221 If **-S** is specified, the message shall be in the following format:

36222 "u=%s,g=%s,o=%s\n", <owner permissions>, <group permissions>,  
36223 <other permissions>

36224 where the three values shall be combinations of letters from the set {r, w, x}; the presence of a  
36225 letter shall indicate that the corresponding bit is clear in the file mode creation mask.

36226 If a *mask* operand is specified, there shall be no output written to standard output.

**36227 STDERR**

36228 The standard error shall be used only for diagnostic messages.

**36229 OUTPUT FILES**

36230 None.

**36231 EXTENDED DESCRIPTION**

36232 None.

**36233 EXIT STATUS**

36234 The following exit values shall be returned:

36235 0 The file mode creation mask was successfully changed, or no *mask* operand was supplied.

36236 >0 An error occurred.

## 36237 CONSEQUENCES OF ERRORS

36238 Default.

## 36239 APPLICATION USAGE

36240 Since *umask* affects the current shell execution environment, it is generally provided as a shell  
36241 regular built-in.36242 In contrast to the negative permission logic provided by the file mode creation mask and the  
36243 octal number form of the *mask* argument, the symbolic form of the *mask* argument specifies those  
36244 permissions that are left alone.

## 36245 EXAMPLES

36246 Either of the commands:

36247 `umask a=rx,ug+w`36248 `umask 002`

36249 sets the mode mask so that subsequently created files have their S\_IWOTH bit cleared.

36250 After setting the mode mask with either of the above commands, the *umask* command can be  
36251 used to write out the current value of the mode mask:36252 `$ umask`36253 `0002`36254 (The output format is unspecified, but historical implementations use the octal integer mode  
36255 format.)36256 `$ umask -S`36257 `u=rwx,g=rwx,o=rx`36258 Either of these outputs can be used as the mask operand to a subsequent invocation of the *umask*  
36259 utility.

36260 Assuming the mode mask is set as above, the command:

36261 `umask g-w`36262 sets the mode mask so that subsequently created files have their S\_IWGRP and S\_IWOTH bits  
36263 cleared.

36264 The command:

36265 `umask -- -w`36266 sets the mode mask so that subsequently created files have all their write bits cleared. Note that  
36267 *mask* operands `-r`, `-w`, `-x` or anything beginning with a hyphen, must be preceded by `--` to  
36268 keep it from being interpreted as an option.

## 36269 RATIONALE

36270 Since *umask* affects the current shell execution environment, it is generally provided as a shell  
36271 regular built-in. If it is called in a subshell or separate utility execution environment, such as one  
36272 of the following:36273 `( umask 002 )`36274 `nohup umask ...`36275 `find . -exec umask ... \;`

36276 it does not affect the file mode creation mask of the environment of the caller.

36277 The description of the historical utility was modified to allow it to use the symbolic modes of  
36278 *chmod*. The `-s` option used in early proposals was changed to `-S` because `-s` could be confused

36279 with a *symbolic\_mode* form of mask referring to the S\_ISUID and S\_ISGID bits.

36280 The default output style is implementation-defined to permit implementors to provide  
36281 migration to the new symbolic style at the time most appropriate to their users. A **-o** flag to  
36282 force octal mode output was omitted because the octal mode may not be sufficient to specify all  
36283 of the information that may be present in the file mode creation mask when more secure file  
36284 access permission checks are implemented.

36285 It has been suggested that trusted systems developers might appreciate ameliorating the  
36286 requirement that the mode mask “affects” the file access permissions, since it seems access  
36287 control lists might replace the mode mask to some degree. The wording has been changed to say  
36288 that it affects the file permission bits, and it leaves the details of the behavior of how they affect  
36289 the file access permissions to the description in the System Interfaces volume of  
36290 IEEE Std 1003.1-2001.

36291 **FUTURE DIRECTIONS**

36292 None.

36293 **SEE ALSO**

36294 Chapter 2 (on page 29), *chmod*, the System Interfaces volume of IEEE Std 1003.1-2001, *umask()*

36295 **CHANGE HISTORY**

36296 First released in Issue 2.

36297 **Issue 6**

36298 The following new requirements on POSIX implementations derive from alignment with the  
36299 Single UNIX Specification:

- 36300 • The octal mode is supported.

**36301 NAME**

36302 unalias — remove alias definitions

**36303 SYNOPSIS**

36304 UP      `unalias alias-name...`

36305      `unalias -a`

36306

**36307 DESCRIPTION**

36308      The *unalias* utility shall remove the definition for each alias name specified. See Section 2.3.1 (on page 32). The aliases shall be removed from the current shell execution environment; see Section 2.12 (on page 61).

**36311 OPTIONS**

36312      The *unalias* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

36314      The following option shall be supported:

36315      **-a**      Remove all alias definitions from the current shell execution environment.

**36316 OPERANDS**

36317      The following operand shall be supported:

36318      *alias-name*      The name of an alias to be removed.

**36319 STDIN**

36320      Not used.

**36321 INPUT FILES**

36322      None.

**36323 ENVIRONMENT VARIABLES**

36324      The following environment variables shall affect the execution of *unalias*:

36325      *LANG*      Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36329      *LC\_ALL*      If set to a non-empty string value, override the values of all the other internationalization variables.

36331      *LC\_CTYPE*      Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

**36334 *LC\_MESSAGES***

36335      Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

36337 XSI      *NLSPATH*      Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**36338 ASYNCHRONOUS EVENTS**

36339      Default.

**36340 STDOUT**

36341      Not used.

**36342 STDERR**

36343 The standard error shall be used only for diagnostic messages.

**36344 OUTPUT FILES**

36345 None.

**36346 EXTENDED DESCRIPTION**

36347 None.

**36348 EXIT STATUS**

36349 The following exit values shall be returned:

36350 0 Successful completion.

36351 >0 One of the *alias-name* operands specified did not represent a valid alias definition, or an  
36352 error occurred.

**36353 CONSEQUENCES OF ERRORS**

36354 Default.

**36355 APPLICATION USAGE**

36356 Since *unalias* affects the current shell execution environment, it is generally provided as a shell  
36357 regular built-in.

**36358 EXAMPLES**

36359 None.

**36360 RATIONALE**

36361 The *unalias* description is based on that from historical KornShell implementations. Known  
36362 differences exist between that and the C shell. The KornShell version was adopted to be  
36363 consistent with all the other KornShell features in this volume of IEEE Std 1003.1-2001, such as  
36364 command line editing.

36365 The **-a** option is the equivalent of the *unalias \** form of the C shell and is provided to address  
36366 security concerns about unknown aliases entering the environment of a user (or application)  
36367 through the allowable implementation-defined predefined alias route or as a result of an *ENV*  
36368 file. (Although *unalias* could be used to simplify the “secure” shell script shown in the *command*  
36369 rationale, it does not obviate the need to quote all command names. An initial call to *unalias -a*  
36370 would have to be quoted in case there was an alias for *unalias*.)

**36371 FUTURE DIRECTIONS**

36372 None.

**36373 SEE ALSO**

36374 Chapter 2 (on page 29), *alias*

**36375 CHANGE HISTORY**

36376 First released in Issue 4.

**36377 Issue 6**

36378 This utility is marked as part of the User Portability Utilities option.

**36379 NAME**

36380       **uname** — return system name

**36381 SYNOPSIS**

36382       **uname** [**-snrvma**]

**36383 DESCRIPTION**

36384       By default, the **uname** utility shall write the operating system name to standard output. When  
36385       options are specified, symbols representing one or more system characteristics shall be written  
36386       to the standard output. The format and contents of the symbols are implementation-defined. On  
36387       systems conforming to the System Interfaces volume of IEEE Std 1003.1-2001, the symbols  
36388       written shall be those supported by the **uname()** function as defined in the System Interfaces  
36389       volume of IEEE Std 1003.1-2001.

**36390 OPTIONS**

36391       The **uname** utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
36392       12.2, Utility Syntax Guidelines.

36393       The following options shall be supported:

- 36394       **-a**       Behave as though all of the options **-mnrv** were specified.
- 36395       **-m**       Write the name of the hardware type on which the system is running to standard  
36396       output.
- 36397       **-n**       Write the name of this node within an implementation-defined communications  
36398       network.
- 36399       **-r**       Write the current release level of the operating system implementation.
- 36400       **-s**       Write the name of the implementation of the operating system.
- 36401       **-v**       Write the current version level of this release of the operating system  
36402       implementation.

36403       If no options are specified, the **uname** utility shall write the operating system name, as if the **-s**  
36404       option had been specified.

**36405 OPERANDS**

36406       None.

**36407 STDIN**

36408       Not used.

**36409 INPUT FILES**

36410       None.

**36411 ENVIRONMENT VARIABLES**

36412       The following environment variables shall affect the execution of **uname**:

- 36413       **LANG**      Provide a default value for the internationalization variables that are unset or null.  
36414                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
36415                  Internationalization Variables for the precedence of internationalization variables  
36416                  used to determine the values of locale categories.)
- 36417       **LC\_ALL**     If set to a non-empty string value, override the values of all the other  
36418                  internationalization variables.
- 36419       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
36420                  characters (for example, single-byte as opposed to multi-byte characters in  
36421                  arguments).

|           |                               |                                                                                                                                                                                                                                                                           |
|-----------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 36422     | <b><i>LC_MESSAGES</i></b>     | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                              |
| 36423     |                               |                                                                                                                                                                                                                                                                           |
| 36424     |                               |                                                                                                                                                                                                                                                                           |
| 36425 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                     |
| 36426     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                           |
| 36427     |                               | Default.                                                                                                                                                                                                                                                                  |
| 36428     | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                           |
| 36429     |                               | By default, the output shall be a single line of the following form:                                                                                                                                                                                                      |
| 36430     |                               | "%s\n", <sysname>                                                                                                                                                                                                                                                         |
| 36431     |                               | If the <b>-a</b> option is specified, the output shall be a single line of the following form:                                                                                                                                                                            |
| 36432     |                               | "%s %s %s %s\n", <sysname>, <nodename>, <release>,                                                                                                                                                                                                                        |
| 36433     |                               | <version>, <machine>                                                                                                                                                                                                                                                      |
| 36434     |                               | Additional implementation-defined symbols may be written; all such symbols shall be written at the end of the line of output before the <newline>.                                                                                                                        |
| 36435     |                               |                                                                                                                                                                                                                                                                           |
| 36436     |                               | If options are specified to select different combinations of the symbols, only those symbols shall be written, in the order shown above for the <b>-a</b> option. If a symbol is not selected for writing, its corresponding trailing <blank>s also shall not be written. |
| 36437     |                               |                                                                                                                                                                                                                                                                           |
| 36438     |                               |                                                                                                                                                                                                                                                                           |
| 36439     | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                           |
| 36440     |                               | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                            |
| 36441     | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                           |
| 36442     |                               | None.                                                                                                                                                                                                                                                                     |
| 36443     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                           |
| 36444     |                               | None.                                                                                                                                                                                                                                                                     |
| 36445     | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                           |
| 36446     |                               | The following exit values shall be returned:                                                                                                                                                                                                                              |
| 36447     |                               | 0 The requested information was successfully written.                                                                                                                                                                                                                     |
| 36448     |                               | >0 An error occurred.                                                                                                                                                                                                                                                     |
| 36449     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                           |
| 36450     |                               | Default.                                                                                                                                                                                                                                                                  |
| 36451     | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                                                                                           |
| 36452     |                               | Note that any of the symbols could include embedded <space>s, which may affect parsing                                                                                                                                                                                    |
| 36453     |                               | algorithms if multiple options are selected for output.                                                                                                                                                                                                                   |
| 36454     |                               | The node name is typically a name that the system uses to identify itself for inter-system                                                                                                                                                                                |
| 36455     |                               | communication addressing.                                                                                                                                                                                                                                                 |
| 36456     | <b>EXAMPLES</b>               |                                                                                                                                                                                                                                                                           |
| 36457     |                               | The following command:                                                                                                                                                                                                                                                    |
| 36458     |                               | uname -sr                                                                                                                                                                                                                                                                 |
| 36459     |                               | writes the operating system name and release level, separated by one or more <blank>s.                                                                                                                                                                                    |

**36460 RATIONALE**

36461 It was suggested that this utility cannot be used portably since the format of the symbols is  
36462 implementation-defined. The POSIX.1 working group could not achieve consensus on defining  
36463 these formats in the underlying *uname()* function, and there was no expectation that this volume  
36464 of IEEE Std 1003.1-2001 would be any more successful. Some applications may still find this  
36465 historical utility of value. For example, the symbols could be used for system log entries or for  
36466 comparison with operator or user input.

**36467 FUTURE DIRECTIONS**

36468 None.

**36469 SEE ALSO**

36470 The System Interfaces volume of IEEE Std 1003.1-2001, *uname()*

**36471 CHANGE HISTORY**

36472 First released in Issue 2.

**36473 NAME**

36474        *uncompress* — expand compressed data

**36475 SYNOPSIS**

36476 XSI     **uncompress [-cfv][file...]**

36477

**36478 DESCRIPTION**

36479        The *uncompress* utility shall restore files to their original state after they have been compressed using the *compress* utility. If no files are specified, the standard input shall be uncompressed to the standard output. If the invoking process has appropriate privileges, the ownership, modes, access time, and modification time of the original file shall be preserved.

36483        This utility shall support the uncompressing of any files produced by the *compress* utility on the same implementation. For files produced by *compress* on other systems, *uncompress* supports 9 to 14-bit compression (see *compress*, **-b**); it is implementation-defined whether values of **-b** greater than 14 are supported.

**36487 OPTIONS**

36488        The *uncompress* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

36490        The following options shall be supported:

36491        **-c**        Write to standard output; no files are changed.

36492        **-f**        Do not prompt for overwriting files. Except when run in the background, if **-f** is not given the user shall be prompted as to whether an existing file should be overwritten. If the standard input is not a terminal and **-f** is not given, *uncompress* shall write a diagnostic message to standard error and exit with a status greater than zero.

36497        **-v**        Write messages to standard error concerning the expansion of each file.

**36498 OPERANDS**

36499        The following operand shall be supported:

36500        *file*       A pathname of a file. If *file* already has the **.Z** suffix specified, it shall be used as the input file and the output file shall be named *file* with the **.Z** suffix removed. Otherwise, *file* shall be used as the name of the output file and *file* with the **.Z** suffix appended shall be used as the input file.

**36504 STDIN**

36505        The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '**-**'.

**36506 INPUT FILES**

36507        Input files shall be in the format produced by the *compress* utility.

**36508 ENVIRONMENT VARIABLES**

36509        The following environment variables shall affect the execution of *uncompress*:

36510        **LANG**      Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36514        **LC\_ALL**     If set to a non-empty string value, override the values of all the other internationalization variables.

- 36516        ***LC\_CTYPE***    Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).
- 36519        ***LC\_MESSAGES***    Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.
- 36522        ***NLSPATH***    Determine the location of message catalogs for the processing of ***LC\_MESSAGES***.
- 36523 **ASYNCHRONOUS EVENTS**
- 36524        Default.
- 36525 **STDOUT**
- 36526        When there are no *file* operands or the **-c** option is specified, the uncompressed output is written to standard output.
- 36528 **STDERR**
- 36529        Prompts shall be written to the standard error output under the conditions specified in the DESCRIPTION and OPTIONS sections. The prompts shall contain the *file* pathname, but their format is otherwise unspecified. Otherwise, the standard error output shall be used only for diagnostic messages.
- 36533 **OUTPUT FILES**
- 36534        Output files are the same as the respective input files to *compress*.
- 36535 **EXTENDED DESCRIPTION**
- 36536        None.
- 36537 **EXIT STATUS**
- 36538        The following exit values shall be returned:
- 36539        0    Successful completion.
- 36540        >0    An error occurred.
- 36541 **CONSEQUENCES OF ERRORS**
- 36542        The input file remains unmodified.
- 36543 **APPLICATION USAGE**
- 36544        The limit of 14 on the *compress -b bits* argument is to achieve portability to all systems (within the restrictions imposed by the lack of an explicit published file format). Some implementations based on 16-bit architectures cannot support 15 or 16-bit uncompression.
- 36547 **EXAMPLES**
- 36548        None.
- 36549 **RATIONALE**
- 36550        None.
- 36551 **FUTURE DIRECTIONS**
- 36552        None.
- 36553 **SEE ALSO**
- 36554        *compress*, *zcat*
- 36555 **CHANGE HISTORY**
- 36556        First released in Issue 4.

**36557 Issue 6**

36558 The normative text is reworded to avoid use of the term “must” for application requirements.

**36559 NAME**

36560        unexpand — convert spaces to tabs

**36561 SYNOPSIS**

36562 UP        **unexpand [-a | -t tablist][file...]**

36563

**36564 DESCRIPTION**

36565        The *unexpand* utility shall copy files or standard input to standard output, converting <blank>s at the beginning of each line into the maximum number of <tab>s followed by the minimum number of <space>s needed to fill the same column positions originally filled by the translated <blank>s. By default, tabstops shall be set at every eighth column position. Each <backspace> shall be copied to the output, and shall cause the column position count for tab calculations to be decremented; the count shall never be decremented to a value less than one.

**36571 OPTIONS**

36572        The *unexpand* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

36574        The following options shall be supported:

36575        **-a**        In addition to translating <blank>s at the beginning of each line, translate all sequences of two or more <blank>s immediately preceding a tab stop to the maximum number of <tab>s followed by the minimum number of <space>s needed to fill the same column positions originally filled by the translated <blank>s.

36580        **-t tablist**        Specify the tab stops. The application shall ensure that the *tablist* option-argument is a single argument consisting of a single positive decimal integer or multiple positive decimal integers, separated by <blank>s or commas, in ascending order. If a single number is given, tabs shall be set *tablist* column positions apart instead of the default 8. If multiple numbers are given, the tabs shall be set at those specific column positions.

36586        The application shall ensure that each tab-stop position *N* is an integer value greater than zero, and the list shall be in strictly ascending order. This is taken to mean that, from the start of a line of output, tabbing to position *N* shall cause the next character output to be in the (*N*+1)th column position on that line. When the **-t** option is not specified, the default shall be the equivalent of specifying **-t 8** (except for the interaction with **-a**, described below).

36592        No <space>-to-<tab> conversions shall occur for characters at positions beyond the last of those specified in a multiple tab-stop list.

36594        When **-t** is specified, the presence or absence of the **-a** option shall be ignored; conversion shall not be limited to the processing of leading <blank>s.

**36596 OPERANDS**

36597        The following operand shall be supported:

36598        *file*        A pathname of a text file to be used as input.

**36599 STDIN**

36600        See the INPUT FILES section.

**36601 INPUT FILES**

36602 The input files shall be text files.

**36603 ENVIRONMENT VARIABLES**

36604 The following environment variables shall affect the execution of *unexpand*:

36605 *LANG* Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36609 *LC\_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

36611 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files), the processing of <tab>s and <space>s, and for the determination of the width in column positions each character would occupy on an output device.

**36616 *LC\_MESSAGES***

36617 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

36619 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**36620 ASYNCHRONOUS EVENTS**

36621 Default.

**36622 STDOUT**

36623 The standard output shall be equivalent to the input files with the specified <space>-to-<tab> conversions.

**36625 STDERR**

36626 The standard error shall be used only for diagnostic messages.

**36627 OUTPUT FILES**

36628 None.

**36629 EXTENDED DESCRIPTION**

36630 None.

**36631 EXIT STATUS**

36632 The following exit values shall be returned:

36633 0 Successful completion.

36634 >0 An error occurred.

**36635 CONSEQUENCES OF ERRORS**

36636 Default.

**36637 APPLICATION USAGE**

36638 One non-intuitive aspect of *unexpand* is its restriction to leading spaces when neither **-a** nor **-t** is specified. Users who always want to convert all spaces in a file can easily alias *unexpand* to use the **-a** or **-t 8** option.

**36641 EXAMPLES**

36642 None.

**36643 RATIONALE**

36644 On several occasions, consideration was given to adding a **-t** option to the *unexpand* utility to complement the **-t** in *expand* (see *expand*). The historical intent of *unexpand* was to translate multiple <blank>s into tab stops, where tab stops were a multiple of eight column positions on most UNIX systems. An early proposal omitted **-t** because it seemed outside the scope of the User Portability Utilities option; it was not described in any of the base documents. However, hard-coding tab stops every eight columns was not suitable for the international community and broke historical precedents for some vendors in the FORTRAN community, so **-t** was restored in conjunction with the list of valid extension categories considered by the standard developers.

36652 Thus, *unexpand* is now the logical converse of *expand*.

**36653 FUTURE DIRECTIONS**

36654 None.

**36655 SEE ALSO**

36656 *expand*, *tabs*

**36657 CHANGE HISTORY**

36658 First released in Issue 4.

**36659 Issue 6**

36660 This utility is marked as part of the User Portability Utilities option.

36661 The definition of the *LC\_CTYPE* environment variable is changed to align with the IEEE P1003.2b draft standard.

36663 The normative text is reworded to avoid use of the term “must” for application requirements.

**36664 NAME**

36665        unget — undo a previous get of an SCCS file (**DEVELOPMENT**)

**36666 SYNOPSIS**

36667 XSI        **unget [-ns][-r SID] file...**

36668

**36669 DESCRIPTION**

36670        The *unget* utility shall reverse the effect of a *get -e* done prior to creating the intended new delta.

**36671 OPTIONS**

36672        The *unget* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

36674        The following options shall be supported:

36675        **-r SID**        Uniquely identify which delta is no longer intended. (This would have been specified by *get* as the new delta.) The use of this option is necessary only if two or more outstanding *get* commands for editing on the same SCCS file were done by the same person (login name).

36679        **-s**        Suppress the writing to standard output of the intended delta's SID.

36680        **-n**        Retain the file that was obtained by *get*, which would normally be removed from the current directory.

**36682 OPERANDS**

36683        The following operands shall be supported:

36684        *file*        A pathname of an existing SCCS file or a directory. If *file* is a directory, the *unget* utility shall behave as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the pathname does not begin with s.) and unreadable files shall be silently ignored.

36688        If exactly one *file* operand appears, and it is '−', the standard input shall be read; each line of the standard input shall be taken to be the name of an SCCS file to be processed. Non-SCCS files and unreadable files shall be silently ignored.

**36691 STDIN**

36692        The standard input shall be a text file used only when the *file* operand is specified as '−'. Each line of the text file shall be interpreted as an SCCS pathname.

**36694 INPUT FILES**

36695        Any SCCS files processed shall be files of an unspecified format.

**36696 ENVIRONMENT VARIABLES**

36697        The following environment variables shall affect the execution of *unget*:

36698        **LANG**        Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

36702        **LC\_ALL**        If set to a non-empty string value, override the values of all the other internationalization variables.

36704        **LC\_CTYPE**        Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

- 36707        ***LC\_MESSAGES***  
36708              Determine the locale that should be used to affect the format and contents of  
36709              diagnostic messages written to standard error.
- 36710        ***NLSPATH***    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 36711 **ASYNCHRONOUS EVENTS**
- 36712              Default.
- 36713 **STDOUT**
- 36714              The standard output shall consist of a line for each file, in the following format:  
36715              "%s\n", <SID removed from file>
- 36716              If there is more than one named file or if a directory or standard input is named, each pathname  
36717              shall be written before each of the preceding lines:
- 36718              "\n%s:\n", <pathname>
- 36719 **STDERR**
- 36720              The standard error shall be used only for diagnostic messages.
- 36721 **OUTPUT FILES**
- 36722              Any SCCS files updated shall be files of an unspecified format. During processing of a *file*, a  
36723              locking *z-file*, as described in *get*, and a *q-file* (a working copy of the *p-file*), may be created and  
36724              deleted. The *p-file* and *g-file*, as described in *get*, shall be deleted.
- 36725 **EXTENDED DESCRIPTION**
- 36726              None.
- 36727 **EXIT STATUS**
- 36728              The following exit values shall be returned:  
36729              0    Successful completion.  
36730              >0   An error occurred.
- 36731 **CONSEQUENCES OF ERRORS**
- 36732              Default.
- 36733 **APPLICATION USAGE**
- 36734              None.
- 36735 **EXAMPLES**
- 36736              None.
- 36737 **RATIONALE**
- 36738              None.
- 36739 **FUTURE DIRECTIONS**
- 36740              None.
- 36741 **SEE ALSO**
- 36742              *delta, get, sact*
- 36743 **CHANGE HISTORY**
- 36744              First released in Issue 2.
- 36745 **Issue 6**
- 36746              The normative text is reworded to avoid use of the term “must” for application requirements.

**36747 NAME**

36748        *uniq* — report or filter out repeated lines in a file

**36749 SYNOPSIS**

36750        *uniq* [-c|-d|-u][-f *fields*][-s *char*][*input\_file* [*output\_file*]]

**36751 DESCRIPTION**

36752        The *uniq* utility shall read an input file comparing adjacent lines, and write one copy of each  
36753        input line on the output. The second and succeeding copies of repeated adjacent input lines shall  
36754        not be written.

36755        Repeated lines in the input shall not be detected if they are not adjacent.

**36756 OPTIONS**

36757        The *uniq* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
36758        12.2, Utility Syntax Guidelines.

36759        The following options shall be supported:

36760        **-c**        Precede each output line with a count of the number of times the line occurred in  
36761        the input.

36762        **-d**        Suppress the writing of lines that are not repeated in the input.

36763        **-f** *fields*    Ignore the first *fields* fields on each input line when doing comparisons, where  
36764        *fields* is a positive decimal integer. A field is the maximal string matched by the  
36765        basic regular expression:

36766        [[:blank:] ]\*[ ^[:blank:] ]\*

36767        If the *fields* option-argument specifies more fields than appear on an input line, a  
36768        null string shall be used for comparison.

36769        **-s** *chars*    Ignore the first *chars* characters when doing comparisons, where *chars* shall be a  
36770        positive decimal integer. If specified in conjunction with the **-f** option, the first  
36771        *chars* characters after the first *fields* fields shall be ignored. If the *chars* option-  
36772        argument specifies more characters than remain on an input line, a null string shall  
36773        be used for comparison.

36774        **-u**        Suppress the writing of lines that are repeated in the input.

**36775 OPERANDS**

36776        The following operands shall be supported:

36777        *input\_file*    A pathname of the input file. If the *input\_file* operand is not specified, or if the  
36778        *input\_file* is ‘-’, the standard input shall be used.

36779        *output\_file*   A pathname of the output file. If the *output\_file* operand is not specified, the  
36780        standard output shall be used. The results are unspecified if the file named by  
36781        *output\_file* is the file named by *input\_file*.

**36782 STDIN**

36783        The standard input shall be used only if no *input\_file* operand is specified or if *input\_file* is ‘-’.  
36784        See the INPUT FILES section.

**36785 INPUT FILES**

36786        The input file shall be a text file.

**36787 ENVIRONMENT VARIABLES**

36788       The following environment variables shall affect the execution of *uniq*:

36789       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
36790                  (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
36791                  Internationalization Variables for the precedence of internationalization variables  
36792                  used to determine the values of locale categories.)

36793       **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
36794                  internationalization variables.

36795       **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
36796                  characters (for example, single-byte as opposed to multi-byte characters in  
36797                  arguments and input files) and which characters constitute a <blank> in the  
36798                  current locale.

36799       **LC\_MESSAGES**

36800                  Determine the locale that should be used to affect the format and contents of  
36801                  diagnostic messages written to standard error.

36802 XSI       **NLSPATH**     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**36803 ASYNCHRONOUS EVENTS**

36804                  Default.

**36805 STDOUT**

36806                  The standard output shall be used only if no *output\_file* operand is specified. See the OUTPUT  
36807                  FILES section.

**36808 STDERR**

36809                  The standard error shall be used only for diagnostic messages.

**36810 OUTPUT FILES**

36811                  If the **-c** option is specified, the application shall ensure that the output file is empty or each line  
36812                  shall be of the form:

36813                  "**%d %s**", <number of duplicates>, <line>

36814                  otherwise, the application shall ensure that the output file is empty or each line shall be of the  
36815                  form:

36816                  "**%s**", <line>

**36817 EXTENDED DESCRIPTION**

36818                  None.

**36819 EXIT STATUS**

36820                  The following exit values shall be returned:

36821                  0    The utility executed successfully.

36822                  >0   An error occurred.

**36823 CONSEQUENCES OF ERRORS**

36824                  Default.

## 36825 APPLICATION USAGE

36826 The *sort* utility can be used to cause repeated lines to be adjacent in the input file.

## 36827 EXAMPLES

36828 The following input file data (but flushed left) was used for a test series on *uniq*:

```
36829 #01 foo0 bar0 fool bar1
36830 #02 bar0 fool bar1 fool
36831 #03 foo0 bar0 fool bar1
36832 #04
36833 #05 foo0 bar0 fool bar1
36834 #06 foo0 bar0 fool bar1
36835 #07 bar0 fool bar1 foo0
```

36836 What follows is a series of test invocations of the *uniq* utility that use a mixture of *uniq* options  
36837 against the input file data. These tests verify the meaning of *adjacent*. The *uniq* utility views the  
36838 input data as a sequence of strings delimited by '\n'. Accordingly, for the *field*th member of  
36839 the sequence, *uniq* interprets unique or repeated adjacent lines strictly relative to the *fields+1*th  
36840 member.

- 36841 1. This first example tests the line counting option, comparing each line of the input file data  
36842 starting from the second field:

```
36843 uniq -c -f 1 uniq_0I.t
36844 1 #01 foo0 bar0 fool bar1
36845 1 #02 bar0 fool bar1 foo0
36846 1 #03 foo0 bar0 fool bar1
36847 1 #04
36848 2 #05 foo0 bar0 fool bar1
36849 1 #07 bar0 fool bar1 foo0
```

36850 The number '2', prefixing the fifth line of output, signifies that the *uniq* utility detected a  
36851 pair of repeated lines. Given the input data, this can only be true when *uniq* is run using  
36852 the **-f 1** option (which shall cause *uniq* to ignore the first field on each input line).

- 36853 2. The second example tests the option to suppress unique lines, comparing each line of the  
36854 input file data starting from the second field:

```
36855 uniq -d -f 1 uniq_0I.t
36856 #05 foo0 bar0 fool bar1
```

- 36857 3. This test suppresses repeated lines, comparing each line of the input file data starting from  
36858 the second field:

```
36859 uniq -u -f 1 uniq_0I.t
36860 #01 foo0 bar0 fool bar1
36861 #02 bar0 fool bar1 foo1
36862 #03 foo0 bar0 fool bar1
36863 #04
36864 #07 bar0 fool bar1 foo0
```

- 36865 4. This suppresses unique lines, comparing each line of the input file data starting from the  
36866 third character:

```
36867 uniq -d -s 2 uniq_0I.t
```

36868 In the last example, the *uniq* utility found no input matching the above criteria.

**36869 RATIONALE**

36870        Some historical implementations have limited lines to be 1080 bytes in length, which does not  
36871        meet the implied {LINE\_MAX} limit.

**36872 FUTURE DIRECTIONS**

36873        None.

**36874 SEE ALSO**

36875        *comm, sort*

**36876 CHANGE HISTORY**

36877        First released in Issue 2.

**36878 Issue 6**

36879        The obsolescent SYNOPSIS and associated text are removed.

36880        The normative text is reworded to avoid use of the term “must” for application requirements.

**36881 NAME**

36882        unlink — call the *unlink()* function

**36883 SYNOPSIS**

36884 XSI        `unlink file`

36885

**36886 DESCRIPTION**

36887        The *unlink* utility shall perform the function call:

36888        `unlink(file);`

36889        A user may need appropriate privilege to invoke the *unlink* utility.

**36890 OPTIONS**

36891        None.

**36892 OPERANDS**

36893        The following operands shall be supported:

36894        *file*        The pathname of an existing file.

**36895 STDIN**

36896        Not used.

**36897 INPUT FILES**

36898        Not used.

**36899 ENVIRONMENT VARIABLES**

36900        The following environment variables shall affect the execution of *unlink*:

36901        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
36902              (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
36903              Internationalization Variables for the precedence of internationalization variables  
36904              used to determine the values of locale categories.)

36905        *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
36906              internationalization variables.

36907        *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
36908              characters (for example, single-byte as opposed to multi-byte characters in  
36909              arguments).

36910        *LC\_MESSAGES*

36911              Determine the locale that should be used to affect the format and contents of  
36912              diagnostic messages written to standard error.

36913        *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**36914 ASYNCHRONOUS EVENTS**

36915        Default.

**36916 STDOUT**

36917        None.

**36918 STDERR**

36919        The standard error shall be used only for diagnostic messages.

**36920 OUTPUT FILES**

36921 None.

**36922 EXTENDED DESCRIPTION**

36923 None.

**36924 EXIT STATUS**

36925 The following exit values shall be returned:

36926 0 Successful completion.

36927 >0 An error occurred.

**36928 CONSEQUENCES OF ERRORS**

36929 Default.

**36930 APPLICATION USAGE**

36931 None.

**36932 EXAMPLES**

36933 None.

**36934 RATIONALE**

36935 None.

**36936 FUTURE DIRECTIONS**

36937 None.

**36938 SEE ALSO**

36939 *link*, *rm*, the System Interfaces volume of IEEE Std 1003.1-2001, *unlink()*

**36940 CHANGE HISTORY**

36941 First released in Issue 5.

## 36942 NAME

36943 uucp — system-to-system copy

## 36944 SYNOPSIS

36945 XSI uucp [-cCdfjmr][-n user] source-file... destination-file

36946

## 36947 DESCRIPTION

36948 The *uucp* utility shall copy files named by the *source-file* argument to the *destination-file* argument. The files named can be on local or remote systems.

36950 The *uucp* utility cannot guarantee support for all character encodings in all circumstances. For  
36951 example, transmission data may be restricted to 7 bits by the underlying network, 8-bit data and  
36952 filenames need not be portable to non-internationalized systems, and so on. Under these  
36953 circumstances, it is recommended that only characters defined in the ISO/IEC 646:1991  
36954 standard International Reference Version (equivalent to ASCII) 7-bit range of characters be used,  
36955 and that only characters defined in the portable filename character set be used for naming files.  
36956 The protocol for transfer of files is unspecified by IEEE Std 1003.1-2001.

36957 Typical implementations of this utility require a communications line configured to use the Base  
36958 Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface, but other  
36959 communications means may be used. On systems where there are no available communications  
36960 means (either temporarily or permanently), this utility shall write an error message describing  
36961 the problem and exit with a non-zero exit status.

## 36962 OPTIONS

36963 The *uucp* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
36964 12.2, Utility Syntax Guidelines.

36965 The following options shall be supported:

- 36966 **-c** Do not copy local file to the spool directory for transfer to the remote machine  
36967 (default).
- 36968 **-C** Force the copy of local files to the spool directory for transfer.
- 36969 **-d** Make all necessary directories for the file copy (default).
- 36970 **-f** Do not make intermediate directories for the file copy.
- 36971 **-j** Write the job identification string to standard output. This job identification can be  
36972 used by *uustat* to obtain the status or terminate a job.
- 36973 **-m** Send mail to the requester when the copy is completed.
- 36974 **-n user** Notify *user* on the remote system that a file was sent.
- 36975 **-r** Do not start the file transfer; just queue the job.

## 36976 OPERANDS

36977 The following operands shall be supported:

36978 *destination-file*, *source-file*

36979 A pathname of a file to be copied to, or from, respectively. Either name can be a  
36980 pathname on the local machine, or can have the form:

36981 *system-name* !*pathname*

36982 where *system-name* is taken from a list of system names that *uucp* knows about.  
36983 The destination *system-name* can also be a list of names such as:

36984                   `system-name ! system-name ! . . . ! system-name ! pathname`

36985                   in which case, an attempt is made to send the file via the specified route to the destination. Care should be taken to ensure that intermediate nodes in the route are willing to forward information.

36988                   The shell pattern matching notation characters '?', '\*', and "[ . . . ]" appearing in *pathname* shall be expanded on the appropriate system.

36990                   Pathnames can be one of:

36991                  1. An absolute pathname.

36992                  2. A pathname preceded by `~user` where *user* is a login name on the specified system and is replaced by that user's login directory. Note that if an invalid login is specified, the default is to the public directory (called *PUBDIR*; the actual location of *PUBDIR* is implementation-defined).

36996                  3. A pathname preceded by `~/destination` where *destination* is appended to *PUBDIR*.

36998                  **Note:** This destination is treated as a filename unless more than one file is being transferred by this request or the destination is already a directory. To ensure that it is a directory, follow the destination with a '/'. For example, `~/dan/` as the destination makes the directory *PUBDIR/dan* if it does not exist and puts the requested files in that directory.

37003                  4. Anything else shall be prefixed by the current directory.

37004                   If the result is an erroneous pathname for the remote system, the copy shall fail. If 37005                   the *destination-file* is a directory, the last part of the *source-file* name shall be used.

37006                   The read, write, and execute permissions given by *uucp* are implementation-defined.

37008 **STDIN**

37009                   Not used.

37010 **INPUT FILES**

37011                   The files to be copied are regular files.

37012 **ENVIRONMENT VARIABLES**

37013                   The following environment variables shall affect the execution of *uucp*:

37014                  **LANG**           Provide a default value for the internationalization variables that are unset or null. 37015                   (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, 37016                   Internationalization Variables for the precedence of internationalization variables 37017                   used to determine the values of locale categories.)

37018                  **LC\_ALL**       If set to a non-empty string value, override the values of all the other 37019                   internationalization variables.

37020                  **LC\_COLLATE**   Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within bracketed filename patterns.

37023                  **LC\_CTYPE**      Determine the locale for the interpretation of sequences of bytes of text data as 37024                   characters (for example, single-byte as opposed to multi-byte characters in 37025                   arguments and input files) and the behavior of character classes within bracketed 37026                   filename patterns (for example, "' [ [ :lower:] ]\*'").

- 37027       ***LC\_MESSAGES***  
37028           Determine the locale that should be used to affect the format and contents of  
37029           diagnostic messages written to standard error, and informative messages written  
37030           to standard output.
- 37031       **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 37032       **ASYNCHRONOUS EVENTS**  
37033           Default.
- 37034       **STDOUT**  
37035           Not used.
- 37036       **STDERR**  
37037           The standard error shall be used only for diagnostic messages.
- 37038       **OUTPUT FILES**  
37039           The output files (which may be on other systems) are copies of the input files.  
37040           If **-m** is used, mail files are modified.
- 37041       **EXTENDED DESCRIPTION**  
37042           None.
- 37043       **EXIT STATUS**  
37044           The following exit values shall be returned:  
37045              0   Successful completion.  
37046              >0   An error occurred.
- 37047       **CONSEQUENCES OF ERRORS**  
37048           Default.
- 37049       **APPLICATION USAGE**  
37050           The domain of remotely accessible files can (and for obvious security reasons usually should) be  
37051           severely restricted.  
  
37052           Note that the '!' character in addresses has to be escaped when using *csh* as a command  
37053           interpreter because of its history substitution syntax. For *ksh* and *sh* the escape is not necessary,  
37054           but may be used.  
  
37055           As noted above, shell metacharacters appearing in pathnames are expanded on the appropriate  
37056           system. On an internationalized system, this is done under the control of local settings of  
37057           *LC\_COLLATE* and *LC\_CTYPE*. Thus, care should be taken when using bracketed filename  
37058           patterns, as collation and typing rules may vary from one system to another. Also be aware that  
37059           certain types of expression (that is, equivalence classes, character classes, and collating symbols)  
37060           need not be supported on non-internationalized systems.
- 37061       **EXAMPLES**  
37062           None.
- 37063       **RATIONALE**  
37064           None.
- 37065       **FUTURE DIRECTIONS**  
37066           None.

37067 **SEE ALSO**37068        *mailx, uuencode, uustat, uux*37069 **CHANGE HISTORY**

37070        First released in Issue 2.

37071 **Issue 6**37072        The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.37073        The UN margin codes and associated shading are removed from the **-C**, **-f**, **-j**, **-n**, and **-r** options in response to The Open Group Base Resolution bwg2001-003.  
37074

**37075 NAME**

37076        *uudecode* — decode a binary file

**37077 SYNOPSIS**

37078 UP        **uudecode [-o *outfile*][*file*]**

37079

**37080 DESCRIPTION**

37081        The *uudecode* utility shall read a file, or standard input if no file is specified, that includes data  
37082        created by the *uuencode* utility. The *uudecode* utility shall scan the input file, searching for data  
37083        compatible with one of the formats specified in *uuencode*, and attempt to create or overwrite the  
37084        file described by the data (or overridden by the **-o** option). The pathname shall be contained in  
37085        the data or specified by the **-o** option. The file access permission bits and contents for the file to  
37086        be produced shall be contained in that data. The mode bits of the created file (other than  
37087        standard output) shall be set from the file access permission bits contained in the data; that is,  
37088        other attributes of the mode, including the file mode creation mask (see *umask*), shall not affect  
37089        the file being produced.

37090        If the pathname of the file to be produced exists, and the user does not have write permission on  
37091        that file, *uudecode* shall terminate with an error. If the pathname of the file to be produced exists,  
37092        and the user has write permission on that file, the existing file shall be overwritten.

37093        If the input data was produced by *uuencode* on a system with a different number of bits per byte  
37094        than on the target system, the results of *uudecode* are unspecified.

**37095 OPTIONS**

37096        The *uudecode* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001,  
37097        Section 12.2, Utility Syntax Guidelines.

37098        The following option shall be supported by the implementation:

37099        **-o *outfile***        A pathname of a file that shall be used instead of any pathname contained in the  
37100        input data. Specifying an *outfile* option-argument of **/dev/stdout** shall indicate  
37101        standard output.

**37102 OPERANDS**

37103        The following operand shall be supported:

37104        *file*        The pathname of a file containing the output of *uuencode*.

**37105 STDIN**

37106        See the INPUT FILES section.

**37107 INPUT FILES**

37108        The input files shall be files containing the output of *uuencode*.

**37109 ENVIRONMENT VARIABLES**

37110        The following environment variables shall affect the execution of *uudecode*:

37111        **LANG**        Provide a default value for the internationalization variables that are unset or null.  
37112        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
37113        Internationalization Variables for the precedence of internationalization variables  
37114        used to determine the values of locale categories.)

37115        **LC\_ALL**        If set to a non-empty string value, override the values of all the other  
37116        internationalization variables.

37117        **LC\_CTYPE**        Determine the locale for the interpretation of sequences of bytes of text data as  
37118        characters (for example, single-byte as opposed to multi-byte characters in  
37119        arguments and input files).

|           |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 37120     | <b>LC_MESSAGES</b>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                 |
| 37123 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                                        |
| 37124     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37125     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 37126     | <b>STDOUT</b>                 | If the file data header encoded by <i>uuencode</i> is – or /dev/stdout, or the –o /dev/stdout option overrides the file data, the standard output shall be in the same format as the file originally encoded by <i>uuencode</i> . Otherwise, the standard output shall not be used.                                                                                                                                                                                          |
| 37130     | <b>STDERR</b>                 | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                               |
| 37132     | <b>OUTPUT FILES</b>           | The output file shall be in the same format as the file originally encoded by <i>uuencode</i> .                                                                                                                                                                                                                                                                                                                                                                              |
| 37134     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37135     |                               | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 37136     | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37137     |                               | The following exit values shall be returned:                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 37138     | 0                             | Successful completion.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 37139     | >0                            | An error occurred.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37140     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37141     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 37142     | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37143     |                               | The user who is invoking <i>uudecode</i> must have write permission on any file being created.                                                                                                                                                                                                                                                                                                                                                                               |
| 37144     |                               | The output of <i>uuencode</i> is essentially an encoded bit stream that is not cognizant of byte boundaries. It is possible that a 9-bit byte target machine can process input from an 8-bit source, if it is aware of the requirement, but the reverse is unlikely to be satisfying. Of course, the only data that is meaningful for such a transfer between architectures is generally character data.                                                                     |
| 37148     | <b>EXAMPLES</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37149     |                               | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 37150     | <b>RATIONALE</b>              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37151     |                               | Input files are not necessarily text files, as stated by an early proposal. Although the <i>uuencode</i> output is a text file, that output could have been wrapped within another file or mail message that is not a text file.                                                                                                                                                                                                                                             |
| 37154     |                               | The –o option is not historical practice, but was added at the request of WG15 so that the user could override the target pathname without having to edit the input data itself.                                                                                                                                                                                                                                                                                             |
| 37156     |                               | In early drafts, the [–o <i>outfile</i> ] option-argument allowed the use of – to mean standard output. The symbol – has only been used previously in IEEE Std 1003.1-2001 as a standard input indicator. The developers of the standard did not wish to overload the meaning of – in this manner. The /dev/stdout concept exists on most modern systems. The /dev/stdout syntax does not refer to a new special file. It is just a magic cookie to specify standard output. |

**37161 FUTURE DIRECTIONS**

37162 None.

**37163 SEE ALSO**

37164 *umask*, *uuencode*

**37165 CHANGE HISTORY**

37166 First released in Issue 4.

**37167 Issue 6**

37168 This utility is marked as part of the User Portability Utilities option.

37169 The **-o** *outfile* option is added, as specified in the IEEE P1003.2b draft standard.

37170 The normative text is reworded to avoid use of the term “must” for application requirements.

**37171 NAME**

37172        *uuencode* — encode a binary file

**37173 SYNOPSIS**

37174 UP        *uuencode [-m][file] decode.pathname*

37175

**37176 DESCRIPTION**

37177        The *uuencode* utility shall write an encoded version of the named input file, or standard input if no *file* is specified, to standard output. The output shall be encoded using one of the algorithms described in the STDOUT section and shall include the file access permission bits (in *chmod* octal or symbolic notation) of the input file and the *decode.pathname*, for re-creation of the file on another system that conforms to this volume of IEEE Std 1003.1-2001.

**37182 OPTIONS**

37183        The *uuencode* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

37185        The following option shall be supported by the implementation:

37186        **-m**        Encode the output using the MIME Base64 algorithm described in STDOUT. If **-m** is not specified, the historical algorithm described in STDOUT shall be used.

**37188 OPERANDS**

37189        The following operands shall be supported:

37190        *decode.pathname*

37191        The pathname of the file into which the *uudecode* utility shall place the decoded file. Specifying a *decode.pathname* operand of */dev/stdout* shall indicate that *uudecode* is to use standard output. If there are characters in *decode.pathname* that are not in the portable filename character set the results are unspecified.

37195        *file*        A pathname of the file to be encoded.

**37196 STDIN**

37197        See the INPUT FILES section.

**37198 INPUT FILES**

37199        Input files can be files of any type.

**37200 ENVIRONMENT VARIABLES**

37201        The following environment variables shall affect the execution of *uuencode*:

37202        *LANG*        Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

37206        *LC\_ALL*      If set to a non-empty string value, override the values of all the other internationalization variables.

37208        *LC\_CTYPE*    Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

37211        *LC\_MESSAGES*

37212        Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

37214 XSI      **NLSPATH**    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

37215 **ASYNCHRONOUS EVENTS**

37216      Default.

37217 **STDOUT**

37218      **uuencode Base64 Algorithm**

37219      The standard output shall be a text file (encoded in the character set of the current locale) that  
37220 begins with the line:

37221      "begin-base64Δ%sΔ%s\n" , <mode> , <decode.pathname>

37222      and ends with the line:

37223      "====\n"

37224      In both cases, the lines shall have no preceding or trailing <blank>s.

37225      The encoding process represents 24-bit groups of input bits as output strings of four encoded  
37226 characters. Proceeding from left to right, a 24-bit input group shall be formed by concatenating  
37227 three 8-bit input groups. Each 24-bit input group then shall be treated as four concatenated 6-bit  
37228 groups, each of which shall be translated into a single digit in the Base64 alphabet. When  
37229 encoding a bit stream via the Base64 encoding, the bit stream shall be presumed to be ordered  
37230 with the most-significant bit first. That is, the first bit in the stream shall be the high-order bit in  
37231 the first byte, and the eighth bit shall be the low-order bit in the first byte, and so on. Each 6-bit  
37232 group is used as an index into an array of 64 printable characters, as shown in Table 4-21.

37233      **Table 4-21 uuencode Base64 Values**

| Value | Encoding | Value | Encoding | Value | Encoding | Value | Encoding |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 0     | A        | 17    | R        | 34    | i        | 51    | z        |
| 1     | B        | 18    | S        | 35    | j        | 52    | 0        |
| 2     | C        | 19    | T        | 36    | k        | 53    | 1        |
| 3     | D        | 20    | U        | 37    | l        | 54    | 2        |
| 4     | E        | 21    | V        | 38    | m        | 55    | 3        |
| 5     | F        | 22    | W        | 39    | n        | 56    | 4        |
| 6     | G        | 23    | X        | 40    | o        | 57    | 5        |
| 7     | H        | 24    | Y        | 41    | p        | 58    | 6        |
| 8     | I        | 25    | Z        | 42    | q        | 59    | 7        |
| 9     | J        | 26    | a        | 43    | r        | 60    | 8        |
| 10    | K        | 27    | b        | 44    | s        | 61    | 9        |
| 11    | L        | 28    | c        | 45    | t        | 62    | +        |
| 12    | M        | 29    | d        | 46    | u        | 63    | /        |
| 13    | N        | 30    | e        | 47    | v        |       |          |
| 14    | O        | 31    | f        | 48    | w        | (pad) | =        |
| 15    | P        | 32    | g        | 49    | x        |       |          |
| 16    | Q        | 33    | h        | 50    | y        |       |          |

37252      The character referenced by the index shall be placed in the output string.

37253      The output stream (encoded bytes) shall be represented in lines of no more than 76 characters  
37254 each. All line breaks or other characters not found in the table shall be ignored by decoding  
37255 software (see *uudecode*).

37256      Special processing shall be performed if fewer than 24 bits are available at the end of a message  
37257 or encapsulated part of a message. A full encoding quantum shall always be completed at the

37258 end of a message. When fewer than 24 input bits are available in an input group, zero bits shall  
 37259 be added (on the right) to form an integral number of 6-bit groups. Output character positions  
 37260 that are not required to represent actual input data shall be set to the character '='. Since all  
 37261 Base64 input is an integral number of octets, only the following cases can arise:

- 37262 1. The final quantum of encoding input is an integral multiple of 24 bits; here, the final unit of  
 37263 encoded output shall be an integral multiple of 4 characters with no '=' padding.
- 37264 2. The final quantum of encoding input is exactly 16 bits; here, the final unit of encoded  
 37265 output shall be three characters followed by one '=' padding character.
- 37266 3. The final quantum of encoding input is exactly 8 bits; here, the final unit of encoded output  
 37267 shall be two characters followed by two '=' padding characters.

37268 A terminating "====" evaluates to nothing and denotes the end of the encoded data.

### 37269 uuencode Historical Algorithm

37270 The standard output shall be a text file (encoded in the character set of the current locale) that  
 37271 begins with the line:

37272 "beginΔ%sΔ%s\n" <mode>, <decode\_pathname>

37273 and ends with the line:

37274 "end\n"

37275 In both cases, the lines shall have no preceding or trailing <blank>s.

37276 The algorithm that shall be used for lines in between **begin** and **end** takes three octets as input  
 37277 and writes four characters of output by splitting the input at six-bit intervals into four octets,  
 37278 containing data in the lower six bits only. These octets shall be converted to characters by adding  
 37279 a value of 0x20 to each octet, so that each octet is in the range [0x20,0x5f], and then it shall be  
 37280 assumed to represent a printable character in the ISO/IEC 646:1991 standard encoded character  
 37281 set. It then shall be translated into the corresponding character codes for the codeset in use in the  
 37282 current locale. (For example, the octet 0x41, representing 'A', would be translated to 'A' in the  
 37283 current codeset, such as 0xc1 if it were EBCDIC.)

37284 Where the bits of two octets are combined, the least significant bits of the first octet shall be  
 37285 shifted left and combined with the most significant bits of the second octet shifted right. Thus  
 37286 the three octets *A*, *B*, *C* shall be converted into the four octets:

```
37287 0x20 + ((A >> 2) & 0x3F)
37288 0x20 + (((A << 4) | ((B >> 4) & 0xF)) & 0x3F)
37289 0x20 + (((B << 2) | ((C >> 6) & 0x3)) & 0x3F)
37290 0x20 + ((C) & 0x3F)
```

37291 These octets then shall be translated into the local character set.

37292 Each encoded line contains a length character, equal to the number of characters to be decoded  
 37293 plus 0x20 translated to the local character set as described above, followed by the encoded  
 37294 characters. The maximum number of octets to be encoded on each line shall be 45.

### 37295 STDRERR

37296 The standard error shall be used only for diagnostic messages.

### 37297 OUTPUT FILES

37298 None.

## 37299 EXTENDED DESCRIPTION

37300 None.

## 37301 EXIT STATUS

37302 The following exit values shall be returned:

37303 0 Successful completion.

37304 >0 An error occurred.

## 37305 CONSEQUENCES OF ERRORS

37306 Default.

## 37307 APPLICATION USAGE

37308 The file is expanded by 35 percent (each three octets become four, plus control information) causing it to take longer to transmit.

37310 Since this utility is intended to create files to be used for data interchange between systems with possibly different codesets, and to represent binary data as a text file, the ISO/IEC 646:1991 standard was chosen for a midpoint in the algorithm as a known reference point. The output from *uuencode* is a text file on the local system. If the output were in the ISO/IEC 646:1991 standard codeset, it might not be a text file (at least because the <newline>s might not match), and the goal of creating a text file would be defeated. If this text file was then carried to another machine with the same codeset, it would be perfectly compatible with that system's *uudecode*. If it was transmitted over a mail system or sent to a machine with a different codeset, it is assumed that, as for every other text file, some translation mechanism would convert it (by the time it reached a user on the other system) into an appropriate codeset. This translation only makes sense from the local codeset, not if the file has been put into a ISO/IEC 646:1991 standard representation first. Similarly, files processed by *uuencode* can be placed in *pax* archives, intermixed with other text files in the same codeset.

## 37323 EXAMPLES

37324 None.

## 37325 RATIONALE

37326 A new algorithm was added at the request of the international community to parallel work in RFC 2045 (MIME). As with the historical *uuencode* format, the Base64 Content-Transfer-Encoding is designed to represent arbitrary sequences of octets in a form that is not humanly readable. A 65-character subset of the ISO/IEC 646:1991 standard is used, enabling 6 bits to be represented per printable character. (The extra 65th character, '=' , is used to signify a special processing function.)

37332 This subset has the important property that it is represented identically in all versions of the ISO/IEC 646:1991 standard, including US ASCII, and all characters in the subset are also represented identically in all versions of EBCDIC. The historical *uuencode* algorithm does not share this property, which is the reason that a second algorithm was added to the ISO POSIX-2 standard.

37337 The string "====" was used for the termination instead of the end used in the original format because the latter is a string that could be valid encoded input.

37339 In an early draft, the **-m** option was named **-b** (for Base64), but it was renamed to reflect its relationship to the RFC 2045. A **-u** was also present to invoke the default algorithm, but since this was not historical practice, it was omitted as being unnecessary.

37342 See the RATIONALE section in *uudecode* for the derivation of the **/dev/stdout** symbol.

**37343 FUTURE DIRECTIONS**

37344 None.

**37345 SEE ALSO**

37346 *chmod, mailx, uudecode*

**37347 CHANGE HISTORY**

37348 First released in Issue 4.

**37349 Issue 6**

37350 This utility is marked as part of the User Portability Utilities option.

37351 The Base64 algorithm and the ability to output to **/dev/stdout** are added as specified in the IEEE P1003.2b draft standard.  
37352

## 37353 NAME

37354        *uustat* — uucp status inquiry and job control

## 37355 SYNOPSIS

37356 XSI     **uustat [ -q | -k *jobid* | -r *jobid* ]**

37357        **uustat [ -s *system* ] [ -u *user* ]**

37358

## 37359 DESCRIPTION

37360        The *uustat* utility shall display the status of, or cancel, previously specified *uucp* requests, or  
37361        provide general status on *uucp* connections to other systems.

37362        When no options are given, *uustat* shall write to standard output the status of all *uucp* requests  
37363        issued by the current user.

37364        Typical implementations of this utility require a communications line configured to use the Base  
37365        Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface, but other  
37366        communications means may be used. On systems where there are no available communications  
37367        means (either temporarily or permanently), this utility shall write an error message describing  
37368        the problem and exit with a non-zero exit status.

## 37369 OPTIONS

37370        The *uustat* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
37371        12.2, Utility Syntax Guidelines.

37372        The following options shall be supported:

37373        **-q**        Write the jobs queued for each machine.

37374        **-k *jobid***    Kill the *uucp* request whose job identification is *jobid*. The application shall ensure  
37375        that the killed *uucp* request belongs to the person invoking *uustat* unless that user  
37376        has appropriate privileges.

37377        **-r *jobid***    Rejuvenate *jobid*. The files associated with *jobid* are touched so that their  
37378        modification time is set to the current time. This prevents the cleanup program  
37379        from deleting the job until the jobs modification time reaches the limit imposed by  
37380        the program.

37381        **-s *system***    Write the status of all *uucp* requests for remote system *system*.

37382        **-u *user***      Write the status of all *uucp* requests issued by *user*.

## 37383 OPERANDS

37384        None.

## 37385 STDIN

37386        Not used.

## 37387 INPUT FILES

37388        None.

## 37389 ENVIRONMENT VARIABLES

37390        The following environment variables shall affect the execution of *uustat*:

37391        **LANG**      Provide a default value for the internationalization variables that are unset or null.  
37392        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
37393        Internationalization Variables for the precedence of internationalization variables  
37394        used to determine the values of locale categories.)

|       |                               |                                                                                                                                                                                                         |
|-------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 37395 | <i>LC_ALL</i>                 | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                |
| 37397 | <i>LC_CTYPE</i>               | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                               |
| 37400 | <i>LC_MESSAGES</i>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error, and informative messages written to standard output.                       |
| 37404 | <i>NLSPATH</i>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                   |
| 37405 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                         |
| 37406 |                               | Default.                                                                                                                                                                                                |
| 37407 | <b>STDOUT</b>                 |                                                                                                                                                                                                         |
| 37408 |                               | The standard output shall consist of information about each job selected, in an unspecified format. The information shall include at least the job ID, the user ID or name, and the remote system name. |
| 37411 | <b>STDERR</b>                 |                                                                                                                                                                                                         |
| 37412 |                               | The standard error shall be used only for diagnostic messages.                                                                                                                                          |
| 37413 | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                         |
| 37414 |                               | None.                                                                                                                                                                                                   |
| 37415 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                         |
| 37416 |                               | None.                                                                                                                                                                                                   |
| 37417 | <b>EXIT STATUS</b>            |                                                                                                                                                                                                         |
| 37418 |                               | The following exit values shall be returned:                                                                                                                                                            |
| 37419 |                               | 0 Successful completion.                                                                                                                                                                                |
| 37420 |                               | >0 An error occurred.                                                                                                                                                                                   |
| 37421 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                         |
| 37422 |                               | Default.                                                                                                                                                                                                |
| 37423 | <b>APPLICATION USAGE</b>      |                                                                                                                                                                                                         |
| 37424 |                               | None.                                                                                                                                                                                                   |
| 37425 | <b>EXAMPLES</b>               |                                                                                                                                                                                                         |
| 37426 |                               | None.                                                                                                                                                                                                   |
| 37427 | <b>RATIONALE</b>              |                                                                                                                                                                                                         |
| 37428 |                               | None.                                                                                                                                                                                                   |
| 37429 | <b>FUTURE DIRECTIONS</b>      |                                                                                                                                                                                                         |
| 37430 |                               | None.                                                                                                                                                                                                   |
| 37431 | <b>SEE ALSO</b>               |                                                                                                                                                                                                         |
| 37432 |                               | <i>uucp</i>                                                                                                                                                                                             |
| 37433 | <b>CHANGE HISTORY</b>         |                                                                                                                                                                                                         |
| 37434 |                               | First released in Issue 2.                                                                                                                                                                              |

**37435 Issue 6**

- 37436      The normative text is reworded to avoid use of the term “must” for application requirements.
- 37437      The *LC\_TIME* and *TZ* entries are removed from the ENVIRONMENT VARIABLES section.
- 37438      The UN margin code and associated shading are removed from the **-q** option in response to The Open Group Base Resolution bwg2001-003.
- 37439

## 37440 NAME

37441 uux — remote command execution

## 37442 SYNOPSIS

37443 XSI uux [-np] *command-string*37444 uux [-jnp] *command-string*

37445

## 37446 DESCRIPTION

37447 The *uux* utility shall gather zero or more files from various systems, execute a shell pipeline (see  
37448 Section 2.9 (on page 47)) on a specified system, and then send the standard output of the  
37449 command to a file on a specified system. Only the first command of a pipeline can have a  
37450 *system-name!* prefix. All other commands in the pipeline shall be executed on the system of the  
37451 first command.

37452 The following restrictions are applicable to the shell pipeline processed by *uux*:

- In gathering files from different systems, pathname expansion shall not be performed by *uux*.  
Thus, a request such as:

37455 uux "c99 remsys!~/\* .c"

37456 would attempt to copy the file named literally \*.c to the local system.

- The redirection operators ">>", "<<", ">|", and ">&" shall not be accepted. Any use of  
these redirection operators shall cause this utility to write an error message describing the  
problem and exit with a non-zero exit status.

- The reserved word ! cannot be used at the head of the pipeline to modify the exit status. (See  
the *command-string* operand description below.)

- Alias substitution shall not be performed.

37463 A filename can be specified as for *uucp*; it can be an absolute pathname, a pathname preceded by  
37464 ~*name* (which is replaced by the corresponding login directory), a pathname specified as ~/dest  
37465 (*dest* is prefixed by the public directory called *PUBDIR*; the actual location of *PUBDIR* is  
37466 implementation-defined), or a simple filename (which is prefixed by *uux* with the current  
37467 directory). See *uucp* for the details.

37468 The execution of commands on remote systems shall take place in an execution directory known  
37469 to the *uucp* system. All files required for the execution shall be put into this directory unless they  
37470 already reside on that machine. Therefore, the application shall ensure that non-local filenames  
37471 (without path or machine reference) are unique within the *uux* request.

37472 The *uux* utility shall attempt to get all files to the execution system. For files that are output files,  
37473 the application shall ensure that the filename is escaped using parentheses.

37474 The remote system shall notify the user by mail if the requested command on the remote system  
37475 was disallowed or the files were not accessible. This notification can be turned off by the -n  
37476 option.

37477 Typical implementations of this utility require a communications line configured to use the Base  
37478 Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General Terminal Interface, but other  
37479 communications means may be used. On systems where there are no available communications  
37480 means (either temporarily or permanently), this utility shall write an error message describing  
37481 the problem and exit with a non-zero exit status.

37482 The *uux* utility cannot guarantee support for all character encodings in all circumstances. For  
37483 example, transmission data may be restricted to 7 bits by the underlying network, 8-bit data and

37484 filenames need not be portable to non-internationalized systems, and so on. Under these  
37485 circumstances, it is recommended that only characters defined in the ISO/IEC 646:1991  
37486 standard International Reference Version (equivalent to ASCII) 7-bit range of characters be used  
37487 and that only characters defined in the portable filename character set be used for naming files.

### 37488 OPTIONS

37489 The *uux* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
37490 12.2, Utility Syntax Guidelines.

37491 The following options shall be supported:

- 37492 **-p** Make the standard input to *uux* the standard input to the *command-string*.
- 37493 **-j** Write the job identification string to standard output. This job identification can be  
37494 used by *ustat* to obtain the status or terminate a job.
- 37495 **-n** Do not notify the user if the command fails.

### 37496 OPERANDS

37497 The following operand shall be supported:

- 37498 *command-string*  
37499 A string made up of one or more arguments that are similar to normal command  
37500 arguments, except that the command and any filenames can be prefixed by  
37501 *system-name!*. A null *system-name* shall be interpreted as the local system.

### 37502 STDIN

37503 The standard input shall not be used unless the '**-**' or **-p** option is specified; in those cases, the  
37504 standard input shall be made the standard input of the *command-string*.

### 37505 INPUT FILES

37506 Input files shall be selected according to the contents of *command-string*.

### 37507 ENVIRONMENT VARIABLES

37508 The following environment variables shall affect the execution of *uux*:

- 37509 **LANG** Provide a default value for the internationalization variables that are unset or null.  
37510 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
37511 Internationalization Variables for the precedence of internationalization variables  
37512 used to determine the values of locale categories.)
- 37513 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
37514 internationalization variables.
- 37515 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
37516 characters (for example, single-byte as opposed to multi-byte characters in  
37517 arguments).
- 37518 **LC\_MESSAGES**  
37519 Determine the locale that should be used to affect the format and contents of  
37520 diagnostic messages written to standard error.
- 37521 **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

### 37522 ASYNCHRONOUS EVENTS

37523 Default.

**37524 STDOUT**

37525 The standard output shall not be used unless the **-j** option is specified; in that case, the job  
37526 identification string shall be written to standard output in the following format:

37527 "%s\n" , <jobid>

**37528 STDERR**

37529 The standard error shall be used only for diagnostic messages.

**37530 OUTPUT FILES**

37531 Output files shall be created or written, or both, according to the contents of *command-string*.

37532 If **-n** is not used, mail files shall be modified following any command or file-access failures on  
37533 the remote system.

**37534 EXTENDED DESCRIPTION**

37535 None.

**37536 EXIT STATUS**

37537 The following exit values shall be returned:

37538 0 Successful completion.

37539 >0 An error occurred.

**37540 CONSEQUENCES OF ERRORS**

37541 Default.

**37542 APPLICATION USAGE**

37543 Note that, for security reasons, many installations limit the list of commands executable on  
37544 behalf of an incoming request from *uux*. Many sites permit little more than the receipt of mail  
37545 via *uux*.

37546 Any characters special to the command interpreter should be quoted either by quoting the entire  
37547 *command-string* or quoting the special characters as individual arguments.

37548 As noted in *uucp*, shell pattern matching notation characters appearing in pathnames are  
37549 expanded on the appropriate local system. This is done under the control of local settings of  
37550 *LC\_COLLATE* and *LC\_CTYPE*. Thus, care should be taken when using bracketed filename  
37551 patterns, as collation and typing rules may vary from one system to another. Also be aware that  
37552 certain types of expression (that is, equivalence classes, character classes, and collating symbols)  
37553 need not be supported on non-internationalized systems.

**37554 EXAMPLES**

1. The following command gets **file1** from system **a** and **file2** from system **b**, executes *diff* on  
the local system, and puts the results in **file.diff** in the local *PUBDIR* directory. (*PUBDIR* is  
the *uucp* public directory on the local system.)

37555 *uux " !diff a!/usr/file1 b!/a4/file2 >!~/file.diff "*

2. The following command fails because *uux* places all files copied to a system in the same  
working directory. Although the files **xyz** are from two different systems, their filenames  
are the same and conflict.

37562 *uux " !diff a!/usr1/xyz b!/usr2/xyz >!~/xyz.diff "*

3. The following command succeeds (assuming *diff* is permitted on system **a**) because the file  
local to system **a** is not copied to the working directory, and hence does not conflict with  
the file from system **c**.

37566                uux "a!diff a!/usr/xyz c!/usr/xyz >!~/xyz.diff"

**37567 RATIONALE**

37568                None.

**37569 FUTURE DIRECTIONS**

37570                None.

**37571 SEE ALSO**

37572                Chapter 2 (on page 29), *uucp*, *uuencode*, *uustat*

**37573 CHANGE HISTORY**

37574                First released in Issue 2.

**37575 Issue 6**

37576                The obsolescent SYNOPSIS is removed.

37577                The normative text is reworded to avoid use of the term “must” for application requirements.

37578                The UN margin code and associated shading are removed from the **-j** option in response to The  
37579                Open Group Base Resolution bwg2001-003.

## 37580 NAME

37581        val — validate SCCS files (**DEVELOPMENT**)

## 37582 SYNOPSIS

37583 XSI     val -

37584        val [-s][-m name][-r SID][-y type] file...

37585

## 37586 DESCRIPTION

37587        The *val* utility shall determine whether the specified *file* is an SCCS file meeting the characteristics specified by the options.

## 37589 OPTIONS

37590        The *val* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines, except that the usage of the '-' operand is not strictly as intended by the guidelines (that is, reading options and operands from standard input).

37593        The following options shall be supported:

- |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 37594 <b>-m</b> <i>name</i> | Specify a <i>name</i> , which is compared with the SCCS %M% keyword in <i>file</i> ; see <i>get</i> .                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 37595 <b>-r</b> <i>SID</i>  | Specify a <i>SID</i> (SCCS Identification String), an SCCS delta number. A check shall be made to determine whether the <i>SID</i> is ambiguous (for example, <b>-r 1</b> is ambiguous because it physically does not exist but implies 1.1, 1.2, and so on, which may exist) or invalid (for example, <b>-r 1.0</b> or <b>-r 1.1.0</b> are invalid because neither case can exist as a valid delta number). If the <i>SID</i> is valid and not ambiguous, a check shall be made to determine whether it actually exists. |
| 37601 <b>-s</b>             | Silence the diagnostic message normally written to standard output for any error that is detected while processing each named file on a given command line.                                                                                                                                                                                                                                                                                                                                                               |
| 37603 <b>-y</b> <i>type</i> | Specify a <i>type</i> , which shall be compared with the SCCS %Y% keyword in <i>file</i> ; see <i>get</i> .                                                                                                                                                                                                                                                                                                                                                                                                               |

## 37605 OPERANDS

37606        The following operands shall be supported:

- |                   |                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 37607 <i>file</i> | A pathname of an existing SCCS file. If exactly one <i>file</i> operand appears, and it is '-', the standard input shall be read: each line shall be independently processed as if it were a command line argument list. (However, the line is not subjected to any of the shell word expansions, such as parameter expansion or quote removal.) |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 37611 STDIN

37612        The standard input shall be a text file used only when the *file* operand is specified as '-'.

## 37613 INPUT FILES

37614        Any SCCS files processed shall be files of an unspecified format.

## 37615 ENVIRONMENT VARIABLES

37616        The following environment variables shall affect the execution of *val*:

- |                     |                                                                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 37617 <b>LANG</b>   | Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.) |
| 37621 <b>LC_ALL</b> | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                              |

37623            *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).  
37624  
37625

**37626      *LC\_MESSAGES***  
37627            Determine the locale that should be used to affect the format and contents of  
37628            diagnostic messages written to standard error, and informative messages written  
37629            to standard output.

**37630 NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

## 37631 ASYNCHRONOUS EVENTS

37632 Default.

37633 STDOUT

37634 The standard output shall consist of informative messages about either:

1. Each file processed
  2. Each command line read from standard input

If the standard input is not used, for each *file* operand yielding a discrepancy, the output line shall have the following format:

37639 "%s: %s\n", <pathname>, <unspecified string>

37640 If standard input is used, a line of input shall be written before each of the preceding lines for  
37641 files containing discrepancies:

37642                "%s:\n", <input\_line>

37643 STDERR

37644 Not used.

## 37645 OUTPUT FILES

37646                  None.

**37647 EXTENDED DESCRIPTION**

37648                  None.

## 37649 EXIT STATUS

37650 The 8-bit code returned by *val* shall be a disjunction of the possible errors; that is, it can be  
37651 interpreted as a bit string where set bits are interpreted as follows:

|       |      |   |                                     |
|-------|------|---|-------------------------------------|
| 37652 | 0x80 | = | Missing file argument.              |
| 37653 | 0x40 | = | Unknown or duplicate option.        |
| 37654 | 0x20 | = | Corrupted SCCS file.                |
| 37655 | 0x10 | = | Cannot open file or file not SCCS.  |
| 37656 | 0x08 | = | <i>SID</i> is invalid or ambiguous. |
| 37657 | 0x04 | = | <i>SID</i> does not exist.          |
| 37658 | 0x02 | = | %Y%, -y mismatch.                   |
| 37659 | 0x01 | = | %M%, -m mismatch.                   |

Note that `val` can process two or more files on a given command line and can process multiple command lines (when reading the standard input). In these cases an aggregate code shall be returned: a logical OR of the codes generated for each command line and file processed.

**37663 CONSEQUENCES OF ERRORS**

37664 Default.

**37665 APPLICATION USAGE**

37666 Since the *val* exit status sets the 0x80 bit, shell applications checking "\$?" cannot tell if it  
37667 terminated due to a missing file argument or receipt of a signal.

**37668 EXAMPLES**

37669 In a directory with three SCCS files—**s.x** (of **t** type “text”), **s.y**, and **s.z** (a corrupted file)—the  
37670 following command could produce the output shown:

```
37671 val - <<EOF
37672 -y source s.x
37673 -m y s.y
37674 s.z
37675 EOF
37676
37677 -y source s.x
37678 s.x: %Y%, -y mismatch
37679 s.z
37679 s.z: corrupted SCCS file
```

**37680 RATIONALE**

37681 None.

**37682 FUTURE DIRECTIONS**

37683 None.

**37684 SEE ALSO**

37685 *admin, delta, get, prs*

**37686 CHANGE HISTORY**

37687 First released in Issue 2.

**37688 Issue 6**

37689 The Open Group Corrigendum U025/4 is applied, correcting a typographical error in the EXIT  
37690 STATUS.

**37691 NAME**

37692 vi — screen-oriented (visual) display editor

**37693 SYNOPSIS**

37694 UP vi [-rR][-c command][-t tagstring][-w size][file ...]

37695

**37696 DESCRIPTION**

37697 This utility shall be provided on systems that both support the User Portability Utilities option  
37698 and define the POSIX2\_CHAR\_TERM symbol. On other systems it is optional.

37699 The *vi* (visual) utility is a screen-oriented text editor. Only the open and visual modes of the  
37700 editor are described in IEEE Std 1003.1-2001; see the line editor *ex* for additional editing  
37701 capabilities used in *vi*. The user can switch back and forth between *vi* and *ex* and execute *ex*  
37702 commands from within *vi*.

37703 This reference page uses the term *edit buffer* to describe the current working text. No specific  
37704 implementation is implied by this term. All editing changes are performed on the edit buffer,  
37705 and no changes to it shall affect any file until an editor command writes the file.

37706 When using *vi*, the terminal screen acts as a window into the editing buffer. Changes made to  
37707 the editing buffer shall be reflected in the screen display; the position of the cursor on the screen  
37708 shall indicate the position within the editing buffer.

37709 Certain terminals do not have all the capabilities necessary to support the complete *vi* definition.  
37710 When these commands cannot be supported on such terminals, this condition shall not produce  
37711 an error message such as “not an editor command” or report a syntax error. The implementation  
37712 may either accept the commands and produce results on the screen that are the result of an  
37713 unsuccessful attempt to meet the requirements of this volume of IEEE Std 1003.1-2001 or report  
37714 an error describing the terminal-related deficiency.

**37715 OPTIONS**

37716 The *vi* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2,  
37717 Utility Syntax Guidelines.

37718 The following options shall be supported:

37719 **-c command** See the *ex* command description of the **-c** option.

37720 **-r** See the *ex* command description of the **-r** option.

37721 **-R** See the *ex* command description of the **-R** option.

37722 **-t tagstring** See the *ex* command description of the **-t** option.

37723 **-w size** See the *ex* command description of the **-w** option.

**37724 OPERANDS**

37725 See the OPERANDS section of the *ex* command for a description of the operands supported by  
37726 the *vi* command.

**37727 STDIN**

37728 If standard input is not a terminal device, the results are undefined. The standard input consists  
37729 of a series of commands and input text, as described in the EXTENDED DESCRIPTION section.

37730 If a read from the standard input returns an error, or if the editor detects an end-of-file condition  
37731 from the standard input, it shall be equivalent to a SIGHUP asynchronous event.

**37732 INPUT FILES**

37733 See the INPUT FILES section of the *ex* command for a description of the input files supported by the *vi* command.

**37735 ENVIRONMENT VARIABLES**

37736 See the ENVIRONMENT VARIABLES section of the *ex* command for the environment variables that affect the execution of the *vi* command.

**37738 ASYNCHRONOUS EVENTS**

37739 See the ASYNCHRONOUS EVENTS section of the *ex* for the asynchronous events that affect the execution of the *vi* command.

**37741 STDOUT**

37742 If standard output is not a terminal device, undefined results occur.

37743 Standard output may be used for writing prompts to the user, for informational messages, and for writing lines from the file.

**37745 STDERR**

37746 If standard output is not a terminal device, undefined results occur.

37747 The standard error shall be used only for diagnostic messages.

**37748 OUTPUT FILES**

37749 See the OUTPUT FILES section of the *ex* command for a description of the output files supported by the *vi* command.

**37751 EXTENDED DESCRIPTION**

37752 If the terminal does not have the capabilities necessary to support an unspecified portion of the *vi* definition, implementations shall start initially in *ex* mode or open mode. Otherwise, after initialization, *vi* shall be in command mode; text input mode can be entered by one of several commands used to insert or change text. In text input mode, <ESC> can be used to return to command mode; other uses of <ESC> are described later in this section; see **Terminate Command or Input Mode** (on page 993).

**37758 Initialization in ex and vi**

37759 See **Initialization in ex and vi** (on page 356) for a description of *ex* and *vi* initialization for the *vi* utility.

**37761 Command Descriptions in vi**

37762 The following symbols are used in this reference page to represent arguments to commands.

37763 *buffer* See the description of *buffer* in the EXTENDED DESCRIPTION section of the *ex* utility; see **Command Descriptions in ex** (on page 366).

37765 In open and visual mode, when a command synopsis shows both [*buffer*] and [*count*] preceding the command name, they can be specified in either order.

37766 *count* A positive integer used as an optional argument to most commands, either to give a repeat count or as a size. This argument is optional and shall default to 1 unless otherwise specified.

37770 The Synopsis lines for the *vi* commands <control>-G, <control>-L, <control>-R, <control>-], %, &, ^, D, m, M, Q, u, U, and ZZ do not have *count* as an optional argument. Regardless, it shall not be an error to specify a *count* to these commands, and any specified *count* shall be ignored.

37774    ***motion*** An optional trailing argument used by the **!**, **<**, **>**, **c**, **d**, and **y** commands, which is used  
 37775 to indicate the region of text that shall be affected by the command. The motion can be  
 37776 either one of the command characters repeated or one of several other *vi* commands  
 37777 (listed in the following table). Each of the applicable commands specifies the region of  
 37778 text matched by repeating the command; each command that can be used as a motion  
 37779 command specifies the region of text it affects.

37780    Commands that take *motion* arguments operate on either lines or characters, depending  
 37781 on the circumstances. When operating on lines, all lines that fall partially or wholly  
 37782 within the text region specified for the command shall be affected. When operating on  
 37783 characters, only the exact characters in the specified text region shall be affected. Each  
 37784 motion command specifies this individually.

37785    When commands that may be motion commands are not used as motion commands,  
 37786 they shall set the current position to the current line and column as specified.

37787    The following commands shall be valid cursor motion commands:

|                            |     |    |   |   |
|----------------------------|-----|----|---|---|
| 37788    <apostrophe>      | (   | -  | j | H |
| 37789    <carriage-return> | )   | \$ | k | L |
| 37790    <comma>           | [ [ | %  | l | M |
| 37791    <control>-H       | ] ] | _  | n | N |
| 37792    <control>-N       | {   | ;  | t | T |
| 37793    <control>-P       | }   | ?  | w | W |
| 37794    <grave accent>    | ^   | b  | B |   |
| 37795    <newline>         | +   | e  | E |   |
| 37796    <space>           |     | f  | F |   |
| 37797    <zero>            | /   | h  | G |   |

37798    Any *count* that is specified to a command that has an associated motion command shall  
 37799 be applied to the motion command. If a *count* is applied to both the command and its  
 37800 associated motion command, the effect shall be multiplicative.

37801    The following symbols are used in this section to specify locations in the edit buffer:

37802    ***current character***

37803    The character that is currently indicated by the cursor.

37804    ***end of a line***

37805    The point located between the last non-<newline> (if any) and the terminating  
 37806 <newline> of a line. For an empty line, this location coincides with the beginning of the  
 37807 line.

37808    ***end of the edit buffer***

37809    The location corresponding to the end of the last line in the edit buffer.

37810    The following symbols are used in this section to specify command actions:

37811    ***bigword*** In the POSIX locale, *vi* shall recognize four kinds of *bigwords*:

- 37812    1. A maximal sequence of non-<blank>s preceded and followed by <blank>s or the  
 37813 beginning or end of a line or the edit buffer
- 37814    2. One or more sequential blank lines
- 37815    3. The first character in the edit buffer
- 37816    4. The last non-<newline> in the edit buffer

- 37817        *word*     In the POSIX locale, *vi* shall recognize five kinds of words:
- 37818        1. A maximal sequence of letters, digits, and underscores, delimited at both ends by:
    - 37819            — Characters other than letters, digits, or underscores
    - 37820            — The beginning or end of a line
    - 37821            — The beginning or end of the edit buffer
  - 37822        2. A maximal sequence of characters other than letters, digits, underscores, or  
37823            <blank>s, delimited at both ends by:
    - 37824            — A letter, digit, underscore
    - 37825            — <blank>s
    - 37826            — The beginning or end of a line
    - 37827            — The beginning or end of the edit buffer
  - 37828        3. One or more sequential blank lines
  - 37829        4. The first character in the edit buffer
  - 37830        5. The last non-<newline> in the edit buffer

37831        *section boundary*

37832        A *section boundary* is one of the following:

- 37833        1. A line whose first character is a <form-feed>
- 37834        2. A line whose first character is an open curly brace (' { ')
- 37835        3. A line whose first character is a period and whose second and third characters  
37836            match a two-character pair in the **sections** edit option (see *ed*)
- 37837        4. A line whose first character is a period and whose only other character matches  
37838            the first character of a two-character pair in the **sections** edit option, where the  
37839            second character of the two-character pair is a <space>
- 37840        5. The first line of the edit buffer
- 37841        6. The last line of the edit buffer if the last line of the edit buffer is empty or if it is a  
37842            ]] or } command; otherwise, the last non-<newline> of the last line of the edit  
37843            buffer

37844        *paragraph boundary*

37845        A *paragraph boundary* is one of the following:

- 37846        1. A section boundary
- 37847        2. A line whose first character is a period and whose second and third characters  
37848            match a two-character pair in the **paragraphs** edit option (see *ed*)
- 37849        3. A line whose first character is a period and whose only other character matches  
37850            the first character of a two-character pair in the **paragraphs** edit option, where the  
37851            second character of the two-character pair is a <space>
- 37852        4. One or more sequential blank lines

37853        *remembered search direction*

37854        See the description of *remembered search direction* in *ed*.

37855        *sentence boundary*

37856        A *sentence boundary* is one of the following:

- 37857        1. A paragraph boundary
- 37858        2. The first non-<blank> that occurs after a paragraph boundary
- 37859        3. The first non-<blank> that occurs after a period ('.'), exclamation mark ('!'),  
37860           or question mark ('?'), followed by two <space>s or the end of a line; any  
37861           number of closing parenthesis (')'), closing brackets ('[]'), double quote ('"'),  
37862           or single quote ('''') characters can appear between the punctuation mark and  
37863           the two <space>s or end-of-line

37864        In the remainder of the description of the *vi* utility, the term "buffer line" refers to a line in the  
37865        edit buffer and the term "display line" refers to the line or lines on the display screen used to  
37866        display one buffer line. The term "current line" refers to a specific "buffer line".

37867        If there are display lines on the screen for which there are no corresponding buffer lines because  
37868        they correspond to lines that would be after the end of the file, they shall be displayed as a single  
37869        tilde ('~') character, plus the terminating <newline>.

37870        The last line of the screen shall be used to report errors or display informational messages. It  
37871        shall also be used to display the input for "line-oriented commands" (/, ?, :, and !). When a line-  
37872        oriented command is executed, the editor shall enter text input mode on the last line on the  
37873        screen, using the respective command characters as prompt characters. (In the case of the !  
37874        command, the associated motion shall be entered by the user before the editor enters text input  
37875        mode.) The line entered by the user shall be terminated by a <newline>, a non-<control>-V-  
37876        escaped <carriage-return>, or unescaped <ESC>. It is unspecified if more characters than  
37877        require a display width minus one column number of screen columns can be entered.

37878        If any command is executed that overwrites a portion of the screen other than the last line of the  
37879        screen (for example, the *ex suspend* or ! commands), other than the *ex shell* command, the user  
37880        shall be prompted for a character before the screen is refreshed and the edit session continued.

37881        <tab>s shall take up the number of columns on the screen set by the **tabstop** edit option (see *ed*),  
37882        unless there are less than that number of columns before the display margin that will cause the  
37883        displayed line to be folded; in this case, they shall only take up the number of columns up to that  
37884        boundary.

37885        The cursor shall be placed on the current line and relative to the current column as specified by  
37886        each command described in the following sections.

37887        In open mode, if the current line is not already displayed, then it shall be displayed.

37888        In visual mode, if the current line is not displayed, then the lines that are displayed shall be  
37889        expanded, scrolled, or redrawn to cause an unspecified portion of the current line to be  
37890        displayed. If the screen is redrawn, no more than the number of display lines specified by the  
37891        value of the **window** edit option shall be displayed (unless the current line cannot be completely  
37892        displayed in the number of display lines specified by the **window** edit option) and the current  
37893        line shall be positioned as close to the center of the displayed lines as possible (within the  
37894        constraints imposed by the distance of the line from the beginning or end of the edit buffer). If  
37895        the current line is before the first line in the display and the screen is scrolled, an unspecified  
37896        portion of the current line shall be placed on the first line of the display. If the current line is after  
37897        the last line in the display and the screen is scrolled, an unspecified portion of the current line  
37898        shall be placed on the last line of the display.

37899        In visual mode, if a line from the edit buffer (other than the current line) does not entirely fit into  
37900        the lines at the bottom of the display that are available for its presentation, the editor may

choose not to display any portion of the line. The lines of the display that do not contain text from the edit buffer for this reason shall each consist of a single '@' character.

In visual mode, the editor may choose for unspecified reasons to not update lines in the display to correspond to the underlying edit buffer text. The lines of the display that do not correctly correspond to text from the edit buffer for this reason shall consist of a single '@' character (plus the terminating <newline>), and the <control>-R command shall cause the editor to update the screen to correctly represent the edit buffer.

Open and visual mode commands that set the current column set it to a column position in the display, and not a character position in the line. In this case, however, the column position in the display shall be calculated for an infinite width display; for example, the column related to a character that is part of a line that has been folded onto additional screen lines will be offset from the display line column where the buffer line begins, not from the beginning of a particular display line.

The display cursor column in the display is based on the value of the current column, as follows, with each rule applied in turn:

1. If the current column is after the last display line column used by the displayed line, the display cursor column shall be set to the last display line column occupied by the last non-<newline> in the current line; otherwise, the display cursor column shall be set to the current column.
2. If the character of which some portion is displayed in the display line column specified by the display cursor column requires more than a single display line column:
  - a. If in text input mode, the display cursor column shall be adjusted to the first display line column in which any portion of that character is displayed.
  - b. Otherwise, the display cursor column shall be adjusted to the last display line column in which any portion of that character is displayed.

The current column shall not be changed by these adjustments to the display cursor column.

If an error occurs during the parsing or execution of a vi command:

- The terminal shall be alerted. Execution of the vi command shall stop, and the cursor (for example, the current line and column) shall not be further modified.
- Unless otherwise specified by the following command sections, it is unspecified whether an informational message shall be displayed.
- Any partially entered vi command shall be discarded.
- If the vi command resulted from a map expansion, all characters from that map expansion shall be discarded, except as otherwise specified by the map command (see ed).
- If the vi command resulted from the execution of a buffer, no further commands caused by the execution of the buffer shall be executed.

37937 **Page Backwards**

37938 *Synopsis:* [count] <control>-B

37939 If in open mode, the <control>-B command shall behave identically to the z command.  
37940 Otherwise, if the current line is the first line of the edit buffer, it shall be an error.

37941 If the **window** edit option is less than 3, display a screen where the last line of the display shall  
37942 be some portion of:

37943 (*current first line*) -1

37944 otherwise, display a screen where the first line of the display shall be some portion of:

37945 (*current first line*) - count x ((**window** edit option) -2)

37946 If this calculation would result in a line that is before the first line of the edit buffer, the first line  
37947 of the display shall display some portion of the first line of the edit buffer.

37948 *Current line:* If no lines from the previous display remain on the screen, set to the last line of the  
37949 display; otherwise, set to (*line* – the number of new lines displayed on this screen).

37950 *Current column:* Set to non-<blank>.

37951 **Scroll Forward**

37952 *Synopsis:* [count] <control>-D

37953 If the current line is the last line of the edit buffer, it shall be an error.

37954 If no *count* is specified, *count* shall default to the *count* associated with the previous <control>-D  
37955 or <control>-U command. If there was no previous <control>-D or <control>-U command, *count*  
37956 shall default to the value of the **scroll** edit option.

37957 If in open mode, write lines starting with the line after the current line, until *count* lines or the  
37958 last line of the file have been written.

37959 *Current line:* If the current line + *count* is past the last line of the edit buffer, set to the last line of  
37960 the edit buffer; otherwise, set to the current line + *count*.

37961 *Current column:* Set to non-<blank>.

37962 **Scroll Forward by Line**

37963 *Synopsis:* [count] <control>-E

37964 Display the line count lines after the last line currently displayed.

37965 If the last line of the edit buffer is displayed, it shall be an error. If there is no line *count* lines  
37966 after the last line currently displayed, the last line of the display shall display some portion of  
37967 the last line of the edit buffer.

37968 *Current line:* Unchanged if the previous current character is displayed; otherwise, set to the first  
37969 line displayed.

37970 *Current column:* Unchanged.

37971       **Page Forward**

37972       *Synopsis:*    [*count*] <control>-F

37973       If in open mode, the <control>-F command shall behave identically to the z command.  
37974       Otherwise, if the current line is the last line of the edit buffer, it shall be an error.

37975       If the **window** edit option is less than 3, display a screen where the first line of the display shall  
37976       be some portion of:

37977       (*current last line*) +1

37978       otherwise, display a screen where the first line of the display shall be some portion of:

37979       (*current first line*) + *count* x ((**window** edit option) -2)

37980       If this calculation would result in a line that is after the last line of the edit buffer, the last line of  
37981       the display shall display some portion of the last line of the edit buffer.

37982       *Current line*: If no lines from the previous display remain on the screen, set to the first line of the  
37983       display; otherwise, set to (*line* + the number of new lines displayed on this screen).

37984       *Current column*: Set to non-<blank>.

37985       **Display Information**

37986       *Synopsis:*    <control>-G

37987       This command shall be equivalent to the **ex file** command.

37988       **Move Cursor Backwards**

37989       *Synopsis:*    [*count*] <control>-H

37990       [*count*] h

37991       the current erase character (see stty)

37992       If there are no characters before the current character on the current line, it shall be an error. If  
37993       there are less than *count* previous characters on the current line, *count* shall be adjusted to the  
37994       number of previous characters on the line.

37995       If used as a motion command:

37996       1. The text region shall be from the character before the starting cursor up to and including  
37997       the *count*th character before the starting cursor.

37998       2. Any text copied to a buffer shall be in character mode.

37999       If not used as a motion command:

38000       *Current line*: Unchanged.

38001       *Current column*: Set to (*column* – the number of columns occupied by *count* characters ending  
38002       with the previous current column).

38003      **Move Down**

38004      *Synopsis:*    [count] <newline>  
38005                [count] <control>-J  
38006                [count] <control>-M  
38007                [count] <control>-N  
38008                [count] j  
38009                [count] <carriage-return>  
38010                [count] +

38011      If there are less than *count* lines after the current line in the edit buffer, it shall be an error.

38012      If used as a motion command:

- 38013      1. The text region shall include the starting line and the next *count* – 1 lines.
- 38014      2. Any text copied to a buffer shall be in line mode.

38015      If not used as a motion command:

38016      *Current line:* Set to *current line*+ *count*.

38017      *Current column:* Set to non-<blank> for the <carriage-return>, <control>-M, and + commands;  
38018      otherwise, unchanged.

38019      **Clear and Redisplay**

38020      *Synopsis:*    <control>-L

38021      If in open mode, clear the screen and redisplay the current line. Otherwise, clear and redisplay  
38022      the screen.

38023      *Current line:* Unchanged.

38024      *Current column:* Unchanged.

38025      **Move Up**

38026      *Synopsis:*    [count] <control>-P  
38027                [count] k  
38028                [count] –

38029      If there are less than *count* lines before the current line in the edit buffer, it shall be an error.

38030      If used as a motion command:

- 38031      1. The text region shall include the starting line and the previous *count* lines.
- 38032      2. Any text copied to a buffer shall be in line mode.

38033      If not used as a motion command:

38034      *Current line:* Set to *current line* – *count*.

38035      *Current column:* Set to non-<blank> for the – command; otherwise, unchanged.

38036       **Redraw Screen**

38037       *Synopsis:*    <control>-R

38038       If any lines have been deleted from the display screen and flagged as deleted on the terminal  
38039       using the @ convention (see the beginning of the EXTENDED DESCRIPTION section), they shall  
38040       be redisplayed to match the contents of the edit buffer.

38041       It is unspecified whether lines flagged with @ because they do not fit on the terminal display  
38042       shall be affected.

38043       *Current line:* Unchanged.

38044       *Current column:* Unchanged.

38045       **Scroll Backward**

38046       *Synopsis:*    [*count*] <control>-U

38047       If the current line is the first line of the edit buffer, it shall be an error.

38048       If no *count* is specified, *count* shall default to the *count* associated with the previous <control>-D  
38049       or <control>-U command. If there was no previous <control>-D or <control>-U command, *count*  
38050       shall default to the value of the **scroll** edit option.

38051       *Current line:* If *count* is greater than the current line, set to 1; otherwise, set to the current line –  
38052       *count*.

38053       *Current column:* Set to non-<blank>.

38054       **Scroll Backward by Line**

38055       *Synopsis:*    [*count*] <control>-Y

38056       Display the line *count* lines before the first line currently displayed.

38057       If the current line is the first line of the edit buffer, it shall be an error. If this calculation would  
38058       result in a line that is before the first line of the edit buffer, the first line of the display shall  
38059       display some portion of the first line of the edit buffer.

38060       *Current line:* Unchanged if the previous current character is displayed; otherwise, set to the first  
38061       line displayed.

38062       *Current column:* Unchanged.

38063       **Edit the Alternate File**

38064       *Synopsis:*    <control>-^

38065       This command shall be equivalent to the ex **edit** command, with the alternate pathname as its  
38066       argument.

38067       **Terminate Command or Input Mode**

38068       *Synopsis:*    <ESC>

38069       If a partial vi command (as defined by at least one, non-*count* character) has been entered,  
38070       discard the *count* and the command character(s).

38071       Otherwise, if no command characters have been entered, and the <ESC> was the result of a map  
38072       expansion, the terminal shall be alerted and the <ESC> character shall be discarded, but it shall  
38073       not be an error.

38074      Otherwise, it shall be an error.

38075      *Current line*: Unchanged.

38076      *Current column*: Unchanged.

### 38077      **Search for tagstring**

38078      *Synopsis*:     <control>- ]

38079      If the current character is not a word or <blank>, it shall be an error.

38080      This command shall be equivalent to the **ex tag** command, with the argument to that command  
38081      defined as follows.

38082      If the current character is a <blank>:

38083        1. Skip all <blank>s after the cursor up to the end of the line.

38084        2. If the end of the line is reached, it shall be an error.

38085      Then, the argument to the **ex tag** command shall be the current character and all subsequent  
38086      characters, up to the first non-word character or the end of the line.

### 38087      **Move Cursor Forward**

38088      *Synopsis*:     [*count*] <space>

38089      [*count*] l (ell)

38090      If there are less than *count* non-<newline>s after the cursor on the current line, *count* shall be  
38091      adjusted to the number of non-<newline>s after the cursor on the line.

38092      If used as a motion command:

38093        1. If the current or *countth* character after the cursor is the last non-<newline> in the line, the  
38094            text region shall be comprised of the current character up to and including the last non-  
38095            <newline> in the line. Otherwise, the text region shall be from the current character up to,  
38096            but not including, the *countth* character after the cursor.

38097        2. Any text copied to a buffer shall be in character mode.

38098      If not used as a motion command:

38099      If there are no non-<newline>s after the current character on the current line, it shall be an error.

38100      *Current line*: Unchanged.

38101      *Current column*: Set to the last column that displays any portion of the *countth* character after the  
38102      current character.

### 38103      **Replace Text with Results from Shell Command**

38104      *Synopsis*:     [*count*] ! *motion shell-commands* <newline>

38105      If the motion command is the ! command repeated:

38106        1. If the edit buffer is empty and no *count* was supplied, the command shall be the equivalent  
38107            of the **ex :read !** command, with the text input, and no text shall be copied to any buffer.

38108        2. Otherwise:

38109            a. If there are less than *count* -1 lines after the current line in the edit buffer, it shall be  
38110              an error.

- 38111            b. The text region shall be from the current line up to and including the next *count* – 1  
38112            lines.

38113            Otherwise, the text region shall be the lines in which any character of the text region specified by  
38114            the motion command appear.

38115            Any text copied to a buffer shall be in line mode.

38116            This command shall be equivalent to the *ex!* command for the specified lines.

#### 38117            **Move Cursor to End-of-Line**

38118            *Synopsis:*     [*count*] \$

38119            It shall be an error if there are less than (*count* – 1) lines after the current line in the edit buffer.

38120            If used as a motion command:

- 38121            1. If *count* is 1:

38122                a. It shall be an error if the line is empty.

38123                b. Otherwise, the text region shall consist of all characters from the starting cursor to  
38124                      the last non-<newline> in the line, inclusive, and any text copied to a buffer shall be  
38125                      in character mode.

38126            2. Otherwise, if the starting cursor position is at or before the first non-<blank> in the line,  
38127            the text region shall consist of the current and the next *count* – 1 lines, and any text saved to  
38128            a buffer shall be in line mode.

38129            3. Otherwise, the text region shall consist of all characters from the starting cursor to the last  
38130            non-<newline> in the line that is *count* – 1 lines forward from the current line, and any text  
38131            copied to a buffer shall be in character mode.

38132            If not used as a motion command:

38133            *Current line:* Set to the *current line* + *count* – 1.

38134            *Current column:* The current column is set to the last display line column of the last non-  
38135            <newline> in the line, or column position 1 if the line is empty.

38136            The current column shall be adjusted to be on the last display line column of the last non-  
38137            <newline> of the current line as subsequent commands change the current line, until a  
38138            command changes the current column.

#### 38139            **Move to Matching Character**

38140            *Synopsis:*     %

38141            If the character at the current position is not a parenthesis, bracket, or curly brace, search  
38142            forward in the line to the first one of those characters. If no such character is found, it shall be an  
38143            error.

38144            The matching character shall be the parenthesis, bracket, or curly brace matching the  
38145            parenthesis, bracket, or curly brace, respectively, that was at the current position or that was  
38146            found on the current line.

38147            Matching shall be determined as follows, for an open parenthesis:

- 38148            1. Set a counter to 1.  
38149            2. Search forwards until a parenthesis is found or the end of the edit buffer is reached.

- 38150        3. If the end of the edit buffer is reached, it shall be an error.
- 38151        4. If an open parenthesis is found, increment the counter by 1.
- 38152        5. If a close parenthesis is found, decrement the counter by 1.
- 38153        6. If the counter is zero, the current character is the matching character.
- 38154 Matching for a close parenthesis shall be equivalent, except that the search shall be backwards,
- 38155 from the starting character to the beginning of the buffer, a close parenthesis shall increment the
- 38156 counter by 1, and an open parenthesis shall decrement the counter by 1.
- 38157 Matching for brackets and curly braces shall be equivalent, except that searching shall be done
- 38158 for open and close brackets or open and close curly braces. It is implementation-defined whether
- 38159 other characters are searched for and matched as well.
- 38160 If used as a motion command:
- 38161        1. If the matching cursor was after the starting cursor in the edit buffer, and the starting
- 38162 cursor position was at or before the first non-<blank> non-<newline> in the starting line,
- 38163 and the matching cursor position was at or after the last non-<blank> non-<newline> in
- 38164 the matching line, the text region shall consist of the current line to the matching line,
- 38165 inclusive, and any text copied to a buffer shall be in line mode.
- 38166        2. If the matching cursor was before the starting cursor in the edit buffer, and the starting
- 38167 cursor position was at or after the last non-<blank> non-<newline> in the starting line, and
- 38168 the matching cursor position was at or before the first non-<blank> non-<newline> in the
- 38169 matching line, the text region shall consist of the current line to the matching line,
- 38170 inclusive, and any text copied to a buffer shall be in line mode.
- 38171        3. Otherwise, the text region shall consist of the starting character to the matching character,
- 38172 inclusive, and any text copied to a buffer shall be in character mode.
- 38173 If not used as a motion command:
- 38174        *Current line*: Set to the line where the matching character is located.
- 38175        *Current column*: Set to the last column where any portion of the matching character is displayed.
- 38176 **Repeat Substitution**
- 38177        *Synopsis*:      &
- 38178        Repeat the previous substitution command. This command shall be equivalent to the *ex &*
- 38179 command with the current line as its addresses, and without *options*, *count*, or *flags*.
- 38180 **Return to Previous Context at Beginning of Line**
- 38181        *Synopsis*:      ' *character*
- 38182        It shall be an error if there is no line in the edit buffer marked by *character*.
- 38183 If used as a motion command:
- 38184        1. If the starting cursor is after the marked cursor, then the locations of the starting cursor
- 38185 and the marked cursor in the edit buffer shall be logically swapped.
- 38186        2. The text region shall consist of the starting line up to and including the marked line, and
- 38187 any text copied to a buffer shall be in line mode.
- 38188 If not used as a motion command:

38189        *Current line*: Set to the line referenced by the mark.

38190        *Current column*: Set to non-<blank>.

38191        **Return to Previous Context**

38192        *Synopsis*:     ` character

38193        It shall be an error if the marked line is no longer in the edit buffer. If the marked line no longer  
38194        contains a character in the saved numbered character position, it shall be as if the marked  
38195        position is the first non-<blank>.

38196        If used as a motion command:

- 38197        1. It shall be an error if the marked cursor references the same character in the edit buffer as  
38198        the starting cursor.

- 38199        2. If the starting cursor is after the marked cursor, then the locations of the starting cursor  
38200        and the marked cursor in the edit buffer shall be logically swapped.

- 38201        3. If the starting line is empty or the starting cursor is at or before the first non-<blank> non-  
38202        <newline> of the starting line, and the marked cursor line is empty or the marked cursor  
38203        references the first character of the marked cursor line, the text region shall consist of all  
38204        lines containing characters from the starting cursor to the line before the marked cursor  
38205        line, inclusive, and any text copied to a buffer shall be in line mode.

- 38206        4. Otherwise, if the marked cursor line is empty or the marked cursor references a character  
38207        at or before the first non-<blank> non-<newline> of the marked cursor line, the region of  
38208        text shall be from the starting cursor to the last non-<newline> of the line before the  
38209        marked cursor line, inclusive, and any text copied to a buffer shall be in character mode.

- 38210        5. Otherwise, the region of text shall be from the starting cursor (inclusive), to the marked  
38211        cursor (exclusive), and any text copied to a buffer shall be in character mode.

38212        If not used as a motion command:

38213        *Current line*: Set to the line referenced by the mark.

38214        *Current column*: Set to the last column in which any portion of the character referenced by the  
38215        mark is displayed.

38216        **Return to Previous Section**

38217        *Synopsis*:     [ [

38218        Move the cursor backward through the edit buffer to the first character of the previous section  
38219        boundary, *count* times.

38220        If used as a motion command:

- 38221        1. If the starting cursor was at the first character of the starting line or the starting line was  
38222        empty, and the first character of the boundary was the first character of the boundary line,  
38223        the text region shall consist of the current line up to and including the line where the  
38224        *count*th next boundary starts, and any text copied to a buffer shall be in line mode.

- 38225        2. If the boundary was the last line of the edit buffer or the last non-<newline> of the last line  
38226        of the edit buffer, the text region shall consist of the last character in the edit buffer up to  
38227        and including the starting character, and any text saved to a buffer shall be in character  
38228        mode.

- 38229        3. Otherwise, the text region shall consist of the starting character up to but not including the  
38230            first character in the *count*th next boundary, and any text copied to a buffer shall be in  
38231            character mode.

38232        If not used as a motion command:

38233        *Current line*: Set to the line where the *count*th next boundary in the edit buffer starts.

38234        *Current column*: Set to the last column in which any portion of the first character of the *count*th  
38235            next boundary is displayed, or column position 1 if the line is empty.

### 38236        Move to Next Section

38237        *Synopsis*:     ]]

38238        Move the cursor forward through the edit buffer to the first character of the next section  
38239            boundary, *count* times.

38240        If used as a motion command:

- 38241        1. If the starting cursor was at the first character of the starting line or the starting line was  
38242            empty, and the first character of the boundary was the first character of the boundary line,  
38243            the text region shall consist of the current line up to and including the line where the  
38244            *count*th previous boundary starts, and any text copied to a buffer shall be in line mode.
- 38245        2. If the boundary was the first line of the edit buffer, the text region shall consist of the first  
38246            character in the edit buffer up to but not including the starting character, and any text  
38247            copied to a buffer shall be in character mode.
- 38248        3. Otherwise, the text region shall consist of the first character in the *count*th previous section  
38249            boundary up to but not including the starting character, and any text copied to a buffer  
38250            shall be in character mode.

38251        If not used as a motion command:

38252        *Current line*: Set to the line where the *count*th previous boundary in the edit buffer starts.

38253        *Current column*: Set to the last column in which any portion of the first character of the *count*th  
38254            previous boundary is displayed, or column position 1 if the line is empty.

### 38255        Move to First Non-<blank> Position on Current Line

38256        *Synopsis*:     ^

38257        If used as a motion command:

- 38258        1. If the line has no non-<blank> non-<newline>s, or if the cursor is at the first non-<blank>  
38259            non-<newline> of the line, it shall be an error.
- 38260        2. If the cursor is before the first non-<blank> non-<newline> of the line, the text region shall  
38261            be comprised of the current character, up to, but not including, the first non-<blank> non-  
38262            <newline> of the line.
- 38263        3. If the cursor is after the first non-<blank> non-<newline> of the line, the text region shall  
38264            be from the character before the starting cursor up to and including the first non-<blank>  
38265            non-<newline> of the line.
- 38266        4. Any text copied to a buffer shall be in character mode.

38267        If not used as a motion command:

38268        *Current line*: Unchanged.

38269        *Current column*: Set to non-<blank>.

### 38270        **Current and Line Above**

38271        *Synopsis:*     [*count*] \_

38272        If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an error.

38273        If used as a motion command:

38274        1. If *count* is less than 2, the text region shall be the current line.

38275        2. Otherwise, the text region shall include the starting line and the next *count* – 1 lines.

38276        3. Any text copied to a buffer shall be in line mode.

38277        If not used as a motion command:

38278        *Current line*: Set to current line + *count* – 1.

38279        *Current column*: Set to non-<blank>.

### 38280        **Move Back to Beginning of Sentence**

38281        *Synopsis:*     [*count*] (

38282        Move backward to the beginning of a sentence. This command shall be equivalent to the [[ command, with the exception that sentence boundaries shall be used instead of section boundaries.

### 38285        **Move Forward to Beginning of Sentence**

38286        *Synopsis:*     [*count*] )

38287        Move forward to the beginning of a sentence. This command shall be equivalent to the ]] command, with the exception that sentence boundaries shall be used instead of section boundaries.

### 38290        **Move Back to Preceding Paragraph**

38291        *Synopsis:*     [*count*] {

38292        Move back to the beginning of the preceding paragraph. This command shall be equivalent to the [[ command, with the exception that paragraph boundaries shall be used instead of section boundaries.

### 38295        **Move Forward to Next Paragraph**

38296        *Synopsis:*     [*count*] }

38297        Move forward to the beginning of the next paragraph. This command shall be equivalent to the ]] command, with the exception that paragraph boundaries shall be used instead of section boundaries.

38300 **Move to Specific Column Position**38301 *Synopsis:* [count] |

38302 For the purposes of this command, lines that are too long for the current display and that have  
38303 been folded shall be treated as having a single, 1-based, number of columns.

38304 If there are less than *count* columns in which characters from the current line are displayed on  
38305 the screen, *count* shall be adjusted to be the last column in which any portion of the line is  
38306 displayed on the screen.

38307 If used as a motion command:

- 38308 1. If the line is empty, or the cursor character is the same as the character on the *count*th  
38309 column of the line, it shall be an error.
- 38310 2. If the cursor is before the *count*th column of the line, the text region shall be comprised of  
38311 the current character, up to but not including the character on the *count*th column of the  
38312 line.
- 38313 3. If the cursor is after the *count*th column of the line, the text region shall be from the  
38314 character before the starting cursor up to and including the character on the *count*th  
38315 column of the line.
- 38316 4. Any text copied to a buffer shall be in character mode.

38317 If not used as a motion command:

38318 *Current line:* Unchanged.

38319 *Current column:* Set to the last column in which any portion of the character that is displayed in  
38320 the *count* column of the line is displayed.

38321 **Reverse Find Character**38322 *Synopsis:* [count] ,

38323 If the last F, f, T, or t command was F, f, T, or t, this command shall be equivalent to an f, F, t, or  
38324 T command, respectively, with the specified *count* and the same search character.

38325 If there was no previous F, f, T, or t command, it shall be an error.

38326 **Repeat**38327 *Synopsis:* [count] .

38328 Repeat the last !, <, >, A, C, D, I, J, O, P, R, S, X, Y, a, c, d, i, o, p, r, s, x, y, or ~ command. It shall  
38329 be an error if none of these commands have been executed. Commands (other than commands  
38330 that enter text input mode) executed as a result of map expansions, shall not change the value of  
38331 the last repeatable command.

38332 Repeated commands with associated motion commands shall repeat the motion command as  
38333 well; however, any specified *count* shall replace the *count*(s) that were originally specified to the  
38334 repeated command or its associated motion command.

38335 If the motion component of the repeated command is f, F, t, or T, the repeated command shall  
38336 not set the remembered search character for the ; and , commands.

38337 If the repeated command is p or P, and the buffer associated with that command was a numeric  
38338 buffer named with a number less than 9, the buffer associated with the repeated command shall  
38339 be set to be the buffer named by the name of the previous buffer logically incremented by 1.

38340 If the repeated character is a text input command, the input text associated with that command  
38341 is repeated literally:

- 38342 • Input characters are neither macro or abbreviation-expanded.
- 38343 • Input characters are not interpreted in any special way with the exception that <newline>,  
38344 <carriage-return>, and <control>-T behave as described in **Input Mode Commands in vi** (on  
38345 page 1019).

38346 *Current line*: Set as described for the repeated command.

38347 *Current column*: Set as described for the repeated command.

### 38348 **Find Regular Expression**

38349 *Synopsis*: /

38350 If the input line contains no non-<newline>s, it shall be equivalent to a line containing only the  
38351 last regular expression encountered. The enhanced regular expressions supported by vi are  
38352 described in **Regular Expressions in ex** (on page 389).

38353 Otherwise, the line shall be interpreted as one or more regular expressions, optionally followed  
38354 by an address offset or a vi z command.

38355 If the regular expression is not the last regular expression on the line, or if a line offset or z  
38356 command is specified, the regular expression shall be terminated by an unescaped ' /'  
38357 character, which shall not be used as part of the regular expression. If the regular expression is  
38358 not the first regular expression on the line, it shall be preceded by zero or more <blank>s, a  
38359 semicolon, zero or more <blank>s, and a leading ' /' character, which shall not be interpreted as  
38360 part of the regular expression. It shall be an error to precede any regular expression with any  
38361 characters other than these.

38362 Each search shall begin from the character after the first character of the last match (or, if it is the  
38363 first search, after the cursor). If the **wrapscan** edit option is set, the search shall continue to the  
38364 character before the starting cursor character; otherwise, to the end of the edit buffer. It shall be  
38365 an error if any search fails to find a match, and an informational message to this effect shall be  
38366 displayed.

38367 An optional address offset (see **Addressing in ex** (on page 359)) can be specified after the last  
38368 regular expression by including a trailing ' /' character after the regular expression and  
38369 specifying the address offset. This offset will be from the line containing the match for the last  
38370 regular expression specified. It shall be an error if the line offset would indicate a line address  
38371 less than 1 or greater than the last line in the edit buffer. An address offset of zero shall be  
38372 supported. It shall be an error to follow the address offset with any other characters than  
38373 <blank>s.

38374 If not used as a motion command, an optional z command (see **Redraw Window** (on page 1018))  
38375 can be specified after the last regular expression by including a trailing ' /' character after the  
38376 regular expression, zero or more <blank>s, a 'z', zero or more <blank>s, an optional new  
38377 window edit option value, zero or more <blank>s, and a location character. The effect shall be as  
38378 if the z command was executed after the / command. It shall be an error to follow the z  
38379 command with any other characters than <blank>s.

38380 The remembered search direction shall be set to forward.

38381 If used as a motion command:

- 38382 1. It shall be an error if the last match references the same character in the edit buffer as the  
38383 starting cursor.

2. If any address offset is specified, the last match shall be adjusted by the specified offset as described previously.
  3. If the starting cursor is after the last match, then the locations of the starting cursor and the last match in the edit buffer shall be logically swapped.
  4. If any address offset is specified, the text region shall consist of all lines containing characters from the starting cursor to the last match line, inclusive, and any text copied to a buffer shall be in line mode.
  5. Otherwise, if the starting line is empty or the starting cursor is at or before the first non-`<blank>` non-`<newline>` of the starting line, and the last match line is empty or the last match starts at the first character of the last match line, the text region shall consist of all lines containing characters from the starting cursor to the line before the last match line, inclusive, and any text copied to a buffer shall be in line mode.
  6. Otherwise, if the last match line is empty or the last match begins at a character at or before the first non-`<blank>` non-`<newline>` of the last match line, the region of text shall be from the current cursor to the last non-`<newline>` of the line before the last match line, inclusive, and any text copied to a buffer shall be in character mode.
  7. Otherwise, the region of text shall be from the current cursor (inclusive), to the first character of the last match (exclusive), and any text copied to a buffer shall be in character mode.

If not used as a motion command:

**Current line:** If a match is found, set to the last matched line plus the address offset, if any; otherwise, unchanged.

**Current column:** Set to the last column on which any portion of the first character in the last matched string is displayed, if a match is found; otherwise, unchanged.

## **Move to First Character in Line**

*Synopsis:* 0 (zero)

Move to the first character on the current line. The character '0' shall not be interpreted as a command if it is immediately preceded by a digit.

If used as a motion command:

1. If the cursor character is the first character in the line, it shall be an error.
  2. The text region shall be from the character before the cursor character up to and including the first character in the line.
  3. Any text copied to a buffer shall be in character mode.

If not used as a motion command:

*Current line:* Unchanged.

**Current column:** The last column in which any portion of the first character in the line is displayed, or if the line is empty, unchanged.

38421       **Execute an ex Command**

38422       *Synopsis:*     :

38423       Execute one or more *ex* commands.

38424       If any portion of the screen other than the last line of the screen was overwritten by any *ex*  
38425       command (except **shell**), *vi* shall display a message indicating that it is waiting for an input from  
38426       the user, and shall then read a character. This action may also be taken for other, unspecified  
38427       reasons.

38428       If the next character entered is a ' : ', another *ex* command shall be accepted and executed. Any  
38429       other character shall cause the screen to be refreshed and *vi* shall return to command mode.

38430       *Current line:* As specified for the *ex* command.

38431       *Current column:* As specified for the *ex* command.

38432       **Repeat Find**

38433       *Synopsis:*     [*count*] ;

38434       This command shall be equivalent to the last **F**, **f**, **T**, or **t** command, with the specified *count*, and  
38435       with the same search character used for the last **F**, **f**, **T**, or **t** command. If there was no previous **F**,  
38436       **f**, **T**, or **t** command, it shall be an error.

38437       **Shift Left**

38438       *Synopsis:*     [*count*] < *motion*

38439       If the motion command is the < command repeated:

1. If there are less than *count* - 1 lines after the current line in the edit buffer, it shall be an error.
2. The text region shall be from the current line, up to and including the next *count* - 1 lines.

38443       Shift any line in the text region specified by the *count* and motion command one shiftwidth (see  
38444       the *ex shiftwidth* option) toward the start of the line, as described by the *ex <* command. The  
38445       unshifted lines shall be copied to the unnamed buffer in line mode.

38446       *Current line:* If the motion was from the current cursor position toward the end of the edit  
38447       buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region  
38448       specified by the motion command.

38449       *Current column:* Set to non-<blank>.

38450       **Shift Right**

38451       *Synopsis:*     [*count*] > *motion*

38452       If the motion command is the > command repeated:

1. If there are less than *count* - 1 lines after the current line in the edit buffer, it shall be an error.
2. The text region shall be from the current line, up to and including the next *count* - 1 lines.

38456       Shift any line with characters in the text region specified by the *count* and motion command one  
38457       shiftwidth (see the *ex shiftwidth* option) away from the start of the line, as described by the *ex >*  
38458       command. The unshifted lines shall be copied into the unnamed buffer in line mode.

38459     *Current line*: If the motion was from the current cursor position toward the end of the edit  
38460     buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region  
38461     specified by the motion command.

38462     *Current column*: Set to non-<blank>.

### 38463     **Scan Backwards for Regular Expression**

38464     *Synopsis*:     ?

38465     Scan backwards; the ? command shall be equivalent to the / command (see **Find Regular**  
38466     **Expression** (on page 1001)) with the following exceptions:

- 38467     1. The input prompt shall be a '?'.  
38468     2. Each search shall begin from the character before the first character of the last match (or, if  
38469       it is the first search, the character before the cursor character).  
38470     3. The search direction shall be from the cursor toward the beginning of the edit buffer, and  
38471       the **wrapscan** edit option shall affect whether the search wraps to the end of the edit buffer  
38472       and continues.  
38473     4. The remembered search direction shall be set to backward.

### 38474     **Execute**

38475     *Synopsis*:     @*buffer*

38476     If the *buffer* is specified as @, the last buffer executed shall be used. If no previous buffer has been  
38477     executed, it shall be an error.

38478     Behave as if the contents of the named buffer were entered as standard input. After each line of a  
38479     line-mode buffer, and all but the last line of a character mode buffer, behave as if a <newline>  
38480     were entered as standard input.

38481     If an error occurs during this process, an error message shall be written, and no more characters  
38482     resulting from the execution of this command shall be processed.

38483     If a *count* is specified, behave as if that count were entered as user input before the characters  
38484     from the @ buffer were entered.

38485     *Current line*: As specified for the individual commands.

38486     *Current column*: As specified for the individual commands.

### 38487     **Reverse Case**

38488     *Synopsis*:     [*count*] ~

38489     Reverse the case of the current character and the next *count* -1 characters, such that lowercase  
38490     characters that have uppercase counterparts shall be changed to uppercase characters, and  
38491     uppercase characters that have lowercase counterparts shall be changed to lowercase characters,  
38492     as prescribed by the current locale. No other characters shall be affected by this command.

38493     If there are less than *count* -1 characters after the cursor in the edit buffer, *count* shall be adjusted  
38494     to the number of characters after the cursor in the edit buffer minus 1.

38495     For the purposes of this command, the next character after the last non-<newline> on the line  
38496     shall be the next character in the edit buffer.

38497     *Current line*: Set to the line including the (*count*-1)th character after the cursor.

38498        *Current column*: Set to the last column in which any portion of the (*count*-1)th character after the  
38499 cursor is displayed.

38500        **Append**

38501        *Synopsis:*     [*count*] a

38502        Enter text input mode after the current cursor position. No characters already in the edit buffer  
38503 shall be affected by this command. A *count* shall cause the input text to be appended *count* -1  
38504 more times to the end of the input.

38505        *Current line/column*: As specified for the text input commands (see **Input Mode Commands in vi**  
38506 (on page 1019)).

38507        **Append at End-of-Line**

38508        *Synopsis:*     [*count*] A

38509        This command shall be equivalent to the vi command:

38510        \$ [*count*] a

38511        (see **Append**).

38512        **Move Backward to Preceding Word**

38513        *Synopsis:*     [*count*] b

38514        With the exception that words are used as the delimiter instead of bigwords, this command shall  
38515 be equivalent to the B command.

38516        **Move Backward to Preceding Bigword**

38517        *Synopsis:*     [*count*] B

38518        If the edit buffer is empty or the cursor is on the first character of the edit buffer, it shall be an  
38519 error. If less than *count* bigwords begin between the cursor and the start of the edit buffer, *count*  
38520 shall be adjusted to the number of bigword beginnings between the cursor and the start of the  
38521 edit buffer.

38522        If used as a motion command:

38523        1. The text region shall be from the first character of the *count*th previous bigword beginning  
38524 up to but not including the cursor character.

38525        2. Any text copied to a buffer shall be in character mode.

38526        If not used as a motion command:

38527        *Current line*: Set to the line containing the *current column*.

38528        *Current column*: Set to the last column upon which any part of the first character of the *count*th  
38529 previous bigword is displayed.

38530       **Change**

38531       *Synopsis:*    `[buffer][count] c motion`

38532       If the motion command is the **c** command repeated:

38533       1. The buffer text shall be in line mode.

38534       2. If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an  
38535       error.

38536       3. The text region shall be from the current line up to and including the next *count* – 1 lines.

38537       Otherwise, the buffer text mode and text region shall be as specified by the motion command.

38538       The replaced text shall be copied into *buffer*, if specified, and into the unnamed buffer. If the text  
38539       to be replaced contains characters from more than a single line, or the buffer text is in line mode,  
38540       the replaced text shall be copied into the numeric buffers as well.

38541       If the buffer text is in line mode:

38542       1. Any lines that contain characters in the region shall be deleted, and the editor shall enter  
38543       text input mode at the beginning of a new line which shall replace the first line deleted.

38544       2. If the **autoindent** edit option is set, **autoindent** characters equal to the **autoindent**  
38545       characters on the first line deleted shall be inserted as if entered by the user.

38546       Otherwise, if characters from more than one line are in the region of text:

38547       1. The text shall be deleted.

38548       2. Any text remaining in the last line in the text region shall be appended to the first line in  
38549       the region, and the last line in the region shall be deleted.

38550       3. The editor shall enter text input mode after the last character not deleted from the first line  
38551       in the text region, if any; otherwise, on the first column of the first line in the region.

38552       Otherwise:

38553       1. If the glyph for ‘\$’ is smaller than the region, the end of the region shall be marked with a  
38554       ‘\$’.

38555       2. The editor shall enter text input mode, overwriting the region of text.

38556       *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
38557       (on page 1019)).

38558       **Change to End-of-Line**

38559       *Synopsis:*    `[buffer][count] C`

38560       This command shall be equivalent to the **vi** command:

38561       `[buffer][count] c$`

38562       See the **c** command.

38563       **Delete**

38564       *Synopsis:*    `[buffer][count] d motion`

38565       If the motion command is the **d** command repeated:

38566       1. The buffer text shall be in line mode.

38567       2. If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an  
38568       error.

38569       3. The text region shall be from the current line up to and including the next *count* – 1 lines.

38570       Otherwise, the buffer text mode and text region shall be as specified by the motion command.

38571       If in open mode, and the current line is deleted, and the line remains on the display, an '@'  
38572       character shall be displayed as the first glyph of that line.

38573       Delete the region of text into *buffer*, if specified, and into the unnamed buffer. If the text to be  
38574       deleted contains characters from more than a single line, or the buffer text is in line mode, the  
38575       deleted text shall be copied into the numeric buffers, as well.

38576       *Current line:* Set to the first text region line that appears in the edit buffer, unless that line has  
38577       been deleted, in which case it shall be set to the last line in the edit buffer, or line 1 if the edit  
38578       buffer is empty.

38579       *Current column:*

38580       1. If the line is empty, set to column position 1.

38581       2. Otherwise, if the buffer text is in line mode or the motion was from the cursor toward the  
38582       end of the edit buffer:

38583           a. If a character from the current line is displayed in the current column, set to the last  
38584           column that displays any portion of that character.

38585           b. Otherwise, set to the last column in which any portion of any character in the line is  
38586           displayed.

38587       3. Otherwise, if a character is displayed in the column that began the text region, set to the  
38588       last column that displays any portion of that character.

38589       4. Otherwise, set to the last column in which any portion of any character in the line is  
38590       displayed.

38591       **Delete to End-of-Line**

38592       *Synopsis:*    `[buffer] D`

38593       Delete the text from the current position to the end of the current line; equivalent to the vi  
38594       command:

38595       `[buffer] d$`

38596       **Move to End-of-Word**

38597       *Synopsis:*    [*count*] e

38598       With the exception that words are used instead of bigwords as the delimiter, this command shall  
38599       be equivalent to the E command.

38600       **Move to End-of-Bigword**

38601       *Synopsis:*    [*count*] E

38602       If the edit buffer is empty it shall be an error. If less than *count* bigwords end between the cursor  
38603       and the end of the edit buffer, *count* shall be adjusted to the number of bigword endings between  
38604       the cursor and the end of the edit buffer.

38605       If used as a motion command:

38606       1. The text region shall be from the last character of the *count*th next bigword up to and  
38607       including the cursor character.

38608       2. Any text copied to a buffer shall be in character mode.

38609       If not used as a motion command:

38610       *Current line:* Set to the line containing the current column.

38611       *Current column:* Set to the last column upon which any part of the last character of the *count*th  
38612       next bigword is displayed.

38613       **Find Character in Current Line (Forward)**

38614       *Synopsis:*    [*count*] f *character*

38615       It shall be an error if *count* occurrences of the character do not occur after the cursor in the line.

38616       If used as a motion command:

38617       1. The text range shall be from the cursor character up to and including the *count*th  
38618       occurrence of the specified character after the cursor.

38619       2. Any text copied to a buffer shall be in character mode.

38620       If not used as a motion command:

38621       *Current line:* Unchanged.

38622       *Current column:* Set to the last column in which any portion of the *count*th occurrence of the  
38623       specified character after the cursor appears in the line.

38624       **Find Character in Current Line (Reverse)**

38625       *Synopsis:*    [*count*] F *character*

38626       It shall be an error if *count* occurrences of the character do not occur before the cursor in the line.

38627       If used as a motion command:

38628       1. The text region shall be from the *count*th occurrence of the specified character before the  
38629       cursor, up to, but not including the cursor character.

38630       2. Any text copied to a buffer shall be in character mode.

38631       If not used as a motion command:

- 38632        *Current line*: Unchanged.
- 38633        *Current column*: Set to the last column in which any portion of the *count*th occurrence of the specified character before the cursor appears in the line.
- 38635        **Move to Line**
- 38636        *Synopsis:*     [*count*] G
- 38637        If *count* is not specified, it shall default to the last line of the edit buffer. If *count* is greater than the last line of the edit buffer, it shall be an error.
- 38639        If used as a motion command:
- 38640            1. The text region shall be from the cursor line up to and including the specified line.
  - 38641            2. Any text copied to a buffer shall be in line mode.
- 38642        If not used as a motion command:
- 38643        *Current line*: Set to *count* if *count* is specified; otherwise, the last line.
- 38644        *Current column*: Set to non-<blank>.
- 38645        **Move to Top of Screen**
- 38646        *Synopsis:*     [*count*] H
- 38647        If the beginning of the line *count* greater than the first line of which any portion appears on the display does not exist, it shall be an error.
- 38649        If used as a motion command:
- 38650            1. If in open mode, the text region shall be the current line.
  - 38651            2. Otherwise, the text region shall be from the starting line up to and including (the first line of the display + *count* - 1).
  - 38653            3. Any text copied to a buffer shall be in line mode.
- 38654        If not used as a motion command:
- 38655        If in open mode, this command shall set the current column to non-<blank> and do nothing else.
- 38656        Otherwise, it shall set the current line and current column as follows.
- 38657        *Current line*: Set to (the first line of the display + *count* - 1).
- 38658        *Current column*: Set to non-<blank>.
- 38659        **Insert Before Cursor**
- 38660        *Synopsis:*     [*count*] i
- 38661        Enter text input mode before the current cursor position. No characters already in the edit buffer shall be affected by this command. A *count* shall cause the input text to be appended *count* - 1 more times to the end of the input.
- 38664        *Current line/column*: As specified for the text input commands (see **Input Mode Commands in vi** (on page 1019)).

38666      **Insert at Beginning of Line**

38667      *Synopsis:*    [count] I

38668      This command shall be equivalent to the vi command ^[count]i.

38669      **Join**

38670      *Synopsis:*    [count] J

38671      If the current line is the last line in the edit buffer, it shall be an error.

38672      This command shall be equivalent to the ex join command with no addresses, and an ex command count value of 1 if count was not specified or if a count of 1 was specified, and an ex command count value of count -1 for any other value of count, except that the current line and column shall be set as follows.

38676      *Current line:* Unchanged.

38677      *Current column:* The last column in which any portion of the character following the last character in the initial line is displayed, or the last non-<newline> in the line if no characters were appended.

38680      **Move to Bottom of Screen**

38681      *Synopsis:*    [count] L

38682      If the beginning of the line count less than the last line of which any portion appears on the display does not exist, it shall be an error.

38684      If used as a motion command:

1. If in open mode, the text region shall be the current line.
2. Otherwise, the text region shall include all lines from the starting cursor line to (the last line of the display -(count -1)).
3. Any text copied to a buffer shall be in line mode.

38689      If not used as a motion command:

1. If in open mode, this command shall set the current column to non-<blank> and do nothing else.
2. Otherwise, it shall set the current line and current column as follows.

38693      *Current line:* Set to (the last line of the display -(count -1)).

38694      *Current column:* Set to non-<blank>.

38695      **Mark Position**

38696      *Synopsis:*    m letter

38697      This command shall be equivalent to the ex mark command with the specified character as an argument.

38699       **Move to Middle of Screen**

38700       *Synopsis:*     M

38701       The middle line of the display shall be calculated as follows:

38702       ( the top line of the display ) + ( ( ( number of lines displayed ) +1 ) / 2 ) -1

38703       If used as a motion command:

- 38704       1. If in open mode, the text region shall be the current line.
- 38705       2. Otherwise, the text region shall include all lines from the starting cursor line up to and
- 38706           including the middle line of the display.
- 38707       3. Any text copied to a buffer shall be in line mode.

38708       If not used as a motion command:

38709       If in open mode, this command shall set the current column to non-<blank> and do nothing else.

38710       Otherwise, it shall set the current line and current column as follows.

38711       *Current line:* Set to the middle line of the display.

38712       *Current column:* Set to non-<blank>.

38713       **Repeat Regular Expression Find (Forward)**

38714       *Synopsis:*     n

38715       If the remembered search direction was forward, the n command shall be equivalent to the vi / command with no characters entered by the user. Otherwise, it shall be equivalent to the vi ? command with no characters entered by the user.

38718       If the n command is used as a motion command for the ! command, the editor shall not enter text input mode on the last line on the screen, and shall behave as if the user entered a single '!' character as the text input.

38721       **Repeat Regular Expression Find (Reverse)**

38722       *Synopsis:*     N

38723       Scan for the next match of the last pattern given to / or ?, but in the reverse direction; this is the reverse of n.

38725       If the remembered search direction was forward, the N command shall be equivalent to the vi ? command with no characters entered by the user. Otherwise, it shall be equivalent to the vi / command with no characters entered by the user. If the N command is used as a motion command for the ! command, the editor shall not enter text input mode on the last line on the screen, and shall behave as if the user entered a single ! character as the text input.

38730       **Insert Empty Line Below**

38731       *Synopsis:*     o

38732       Enter text input mode in a new line appended after the current line. A count shall cause the input

38733       text to be appended count -1 more times to the end of the already added text, each time starting

38734       on a new, appended line.

38735       *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi** (on page 1019)).

38737      **Insert Empty Line Above**

38738      *Synopsis:*    `O`

38739      Enter text input mode in a new line inserted before the current line. A *count* shall cause the input  
38740      text to be appended *count* –1 more times to the end of the already added text, each time starting  
38741      on a new, appended line.

38742      *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
38743      (on page 1019)).

38744      **Put from Buffer Following**

38745      *Synopsis:*    `[buffer] p`

38746      If no *buffer* is specified, the unnamed buffer shall be used.

38747      If the buffer text is in line mode, the text shall be appended below the current line, and each line  
38748      of the buffer shall become a new line in the edit buffer. A *count* shall cause the buffer text to be  
38749      appended *count* –1 more times to the end of the already added text, each time starting on a new,  
38750      appended line.

38751      If the buffer text is in character mode, the text shall be appended into the current line after the  
38752      cursor, and each line of the buffer other than the first and last shall become a new line in the edit  
38753      buffer. A *count* shall cause the buffer text to be appended *count* –1 more times to the end of the  
38754      already added text, each time starting after the last added character.

38755      *Current line:* If the buffer text is in line mode, set the line to line +1; otherwise, unchanged.

38756      *Current column:* If the buffer text is in line mode:

1. If there is a non-<blank> in the first line of the buffer, set to the last column on which any  
38758      portion of the first non-<blank> in the line is displayed.
2. If there is no non-<blank> in the first line of the buffer, set to the last column on which any  
38760      portion of the last non-<newline> in the first line of the buffer is displayed.

38761      If the buffer text is in character mode:

1. If the text in the buffer is from more than a single line, then set to the last column on which  
38763      any portion of the first character from the buffer is displayed.
2. Otherwise, if the buffer is the unnamed buffer, set to the last column on which any portion  
38765      of the last character from the buffer is displayed.
3. Otherwise, set to the first column on which any portion of the first character from the  
38767      buffer is displayed.

38768      **Put from Buffer Before**

38769      *Synopsis:*    `[buffer] P`

38770      If no *buffer* is specified, the unnamed buffer shall be used.

38771      If the buffer text is in line mode, the text shall be inserted above the current line, and each line of  
38772      the buffer shall become a new line in the edit buffer. A *count* shall cause the buffer text to be  
38773      appended *count* –1 more times to the end of the already added text, each time starting on a new,  
38774      appended line.

38775      If the buffer text is in character mode, the text shall be inserted into the current line before the  
38776      cursor, and each line of the buffer other than the first and last shall become a new line in the edit  
38777      buffer. A *count* shall cause the buffer text to be appended *count* –1 more times to the end of the

- 38778 already added text, each time starting after the last added character.
- 38779 *Current line*: Unchanged.
- 38780 *Current column*: If the buffer text is in line mode:
- 38781 1. If there is a non-<blank> in the first line of the buffer, set to the last column on which any portion of that character is displayed.
  - 38782 2. If there is no non-<blank> in the first line of the buffer, set to the last column on which any portion of the last non-<newline> in the first line of the buffer is displayed.
- 38783 If the buffer text is in character mode:
- 38784 1. If the buffer is the unnamed buffer, set to the last column on which any portion of the last character from the buffer is displayed.
  - 38785 2. Otherwise, set to the first column on which any portion of the first character from the buffer is displayed.
- 38790 **Enter ex Mode**
- 38791 *Synopsis*: Q
- 38792 Leave visual or open mode and enter *ex* command mode.
- 38793 *Current line*: Unchanged.
- 38794 *Current column*: Unchanged.
- 38795 **Replace Character**
- 38796 *Synopsis*: [count] r character
- 38797 Replace the *count* characters at and after the cursor with the specified character. If there are less than *count* non-<newline>s at and after the cursor on the line, it shall be an error.
- 38798 If character is <control>-V, any next character other than the <newline> shall be stripped of any special meaning and used as a literal character.
- 38801 If character is <ESC>, no replacement shall be made and the current line and current column shall be unchanged.
- 38802 If character is <carriage-return> or <newline>, *count* new lines shall be appended to the current line. All but the last of these lines shall be empty. *count* characters at and after the cursor shall be discarded, and any remaining characters after the cursor in the current line shall be moved to the last of the new lines. If the **autoindent** edit option is set, they shall be preceded by the same number of **autoindent** characters found on the line from which the command was executed.
- 38803 *Current line*: Unchanged unless the replacement character is a <carriage-return> or <newline>, in which case it shall be set to line + *count*.
- 38804 *Current column*: Set to the last column position on which a portion of the last replaced character is displayed, or if the replacement character caused new lines to be created, set to non-<blank>.

38812      **Replace Characters**38813      *Synopsis:*    R

38814      Enter text input mode at the current cursor position possibly replacing text on the current line. A  
38815      *count* shall cause the input text to be appended *count* – 1 more times to the end of the input.

38816      *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
38817      (on page 1019)).

38818      **Substitute Character**38819      *Synopsis:*    [buffer][count] s

38820      This command shall be equivalent to the vi command:

38821      [buffer][count] c&lt;space&gt;

38822      **Substitute Lines**38823      *Synopsis:*    [buffer][count] S

38824      This command shall be equivalent to the vi command:

38825      [buffer][count] c\_

38826      **Move Cursor to Before Character (Forward)**38827      *Synopsis:*    [count] t character

38828      It shall be an error if *count* occurrences of the character do not occur after the cursor in the line.

38829      If used as a motion command:

- 38830      1. The text region shall be from the cursor up to but not including the *count*th occurrence of  
38831      the specified character after the cursor.

- 38832      2. Any text copied to a buffer shall be in character mode.

38833      If not used as a motion command:

38834      *Current line:* Unchanged.

38835      *Current column:* Set to the last column in which any portion of the character before the *count*th  
38836      occurrence of the specified character after the cursor appears in the line.

38837      **Move Cursor to After Character (Reverse)**38838      *Synopsis:*    [count] T character

38839      It shall be an error if *count* occurrences of the character do not occur before the cursor in the line.

38840      If used as a motion command:

- 38841      1. If the character before the cursor is the specified character, it shall be an error.
- 38842      2. The text region shall be from the character before the cursor up to but not including the  
38843      *count*th occurrence of the specified character before the cursor.

- 38844      3. Any text copied to a buffer shall be in character mode.

38845      If not used as a motion command:

38846        *Current line*: Unchanged.  
38847        *Current column*: Set to the last column in which any portion of the character after the *count*th  
38848        occurrence of the specified character before the cursor appears in the line.

### 38849        Undo

38850        *Synopsis*:        u

38851        This command shall be equivalent to the *ex undo* command except that the current line and  
38852        current column shall be set as follows:

38853        *Current line*: Set to the first line added or changed if any; otherwise, move to the line preceding  
38854        any deleted text if one exists; otherwise, move to line 1.

38855        *Current column*: If undoing an *ex* command, set to the first non-<blank>.

38856        Otherwise, if undoing a text input command:

- 38857        1. If the command was a C, c, O, o, R, S, or s command, the current column shall be set to the  
38858        value it held when the text input command was entered.
- 38859        2. Otherwise, set to the last column in which any portion of the first character after the  
38860        deleted text is displayed, or, if no non-<newline>s follow the text deleted from this line, set  
38861        to the last column in which any portion of the last non-<newline> in the line is displayed,  
38862        or 1 if the line is empty.

38863        Otherwise, if a single line was modified (that is, not added or deleted) by the u command:

- 38864        1. If text was added or changed, set to the last column in which any portion of the first  
38865        character added or changed is displayed.
- 38866        2. If text was deleted, set to the last column in which any portion of the first character after  
38867        the deleted text is displayed, or, if no non-<newline>s follow the deleted text, set to the last  
38868        column in which any portion of the last non-<newline> in the line is displayed, or 1 if the  
38869        line is empty.

38870        Otherwise, set to non-<blank>.

### 38871        Undo Current Line

38872        *Synopsis*:        U

38873        Restore the current line to its state immediately before the most recent time that it became the  
38874        current line.

38875        *Current line*: Unchanged.

38876        *Current column*: Set to the first column in the line in which any portion of the first character in  
38877        the line is displayed.

### 38878        Move to Beginning of Word

38879        *Synopsis*:        [count] w

38880        With the exception that words are used as the delimiter instead of bigwords, this command shall  
38881        be equivalent to the W command.

38882      **Move to Beginning of Bigword**38883      *Synopsis:*    [count] w

38884      If the edit buffer is empty, it shall be an error. If there are less than *count* bigwords between the  
38885      cursor and the end of the edit buffer, *count* shall be adjusted to move the cursor to the last  
38886      bigword in the edit buffer.

38887      If used as a motion command:

- 38888      1. If the associated command is c, *count* is 1, and the cursor is on a <blank>, the region of text  
38889      shall be the current character and no further action shall be taken.
- 38890      2. If there are less than *count* bigwords between the cursor and the end of the edit buffer, then  
38891      the command shall succeed, and the region of text shall include the last character of the  
38892      edit buffer.
- 38893      3. If there are <blank>s or an end-of-line that precede the *count*th bigword, and the associated  
38894      command is c, the region of text shall be up to and including the last character before the  
38895      preceding <blank>s or end-of-line.
- 38896      4. If there are <blank>s or an end-of-line that precede the bigword, and the associated  
38897      command is d or y, the region of text shall be up to and including the last <blank> before  
38898      the start of the bigword or end-of-line.
- 38899      5. Any text copied to a buffer shall be in character mode.

38900      If not used as a motion command:

- 38901      1. If the cursor is on the last character of the edit buffer, it shall be an error.

38902      *Current line:* Set to the line containing the current column.

38903      *Current column:* Set to the last column in which any part of the first character of the *count*th next  
38904      bigword is displayed.

38905      **Delete Character at Cursor**38906      *Synopsis:*    [buffer][count] x

38907      Delete the *count* characters at and after the current character into *buffer*, if specified, and into the  
38908      unnamed buffer.

38909      If the line is empty, it shall be an error. If there are less than *count* non-<newline>s at and after  
38910      the cursor on the current line, *count* shall be adjusted to the number of non-<newline>s at and  
38911      after the cursor.

38912      *Current line:* Unchanged.

38913      *Current column:* If the line is empty, set to column position 1. Otherwise, if there were *count* or  
38914      less non-<newline>s at and after the cursor on the current line, set to the last column that  
38915      displays any part of the last non-<newline> of the line. Otherwise, unchanged.

38916       **Delete Character Before Cursor**

38917       *Synopsis:*    `[buffer][count] X`

38918       Delete the *count* characters before the current character into *buffer*, if specified, and into the  
38919       unnamed buffer.

38920       If there are no characters before the current character on the current line, it shall be an error. If  
38921       there are less than *count* previous characters on the current line, *count* shall be adjusted to the  
38922       number of previous characters on the line.

38923       *Current line:* Unchanged.

38924       *Current column:* Set to (current column – the width of the deleted characters).

38925       **Yank**

38926       *Synopsis:*    `[buffer][count] y motion`

38927       Copy (yank) the region of text into *buffer*, if specified, and into the unnamed buffer.

38928       If the motion command is the *y* command repeated:

- 38929       1. The buffer shall be in line mode.
- 38930       2. If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an  
38931       error.
- 38932       3. The text region shall be from the current line up to and including the next *count* – 1 lines.

38933       Otherwise, the buffer text mode and text region shall be as specified by the motion command.

38934       *Current line:* If the motion was from the current cursor position toward the end of the edit  
38935       buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region  
38936       specified by the motion command.

38937       *Current column:*

- 38938       1. If the motion was from the current cursor position toward the end of the edit buffer,  
38939       unchanged.
- 38940       2. Otherwise, if the current line is empty, set to column position 1.
- 38941       3. Otherwise, set to the last column that displays any part of the first character in the file that  
38942       is part of the text region specified by the motion command.

38943       **Yank Current Line**

38944       *Synopsis:*    `[buffer][count] Y`

38945       This command shall be equivalent to the *vi* command:

38946       `[buffer][count] y_`

38947       **Redraw Window**

38948       If in open mode, the **z** command shall have the Synopsis:

38949       *Synopsis:*     [*count*] **z**

38950       If *count* is not specified, it shall default to the **window** edit option -1. The **z** command shall be equivalent to the **ex z** command, with a type character of = and a *count* of *count*-2, except that the current line and current column shall be set as follows, and the **window** edit option shall not be affected. If the calculation for the *count* argument would result in a negative number, the *count* argument to the **ex z** command shall be zero. A blank line shall be written after the last line is written.

38956       *Current line:* Unchanged.

38957       *Current column:* Unchanged.

38958       If not in open mode, the **z** command shall have the following Synopsis:

38959       *Synopsis:*     [*line*] **z** [*count*] *character*

38960       If *line* is not specified, it shall default to the current line. If *line* is specified, but is greater than the number of lines in the edit buffer, it shall default to the number of lines in the edit buffer.

38962       If *count* is specified, the value of the **window** edit option shall be set to *count* (as described in the **ex window** command), and the screen shall be redrawn.

38964       *line* shall be placed as specified by the following characters:

38965       <newline>, <carriage-return>

38966       Place the beginning of the line on the first line of the display.

38967       · Place the beginning of the line in the center of the display. The middle line of the display shall be calculated as described for the **M** command.

38969       – Place an unspecified portion of the line on the last line of the display.

38970       + If *line* was specified, equivalent to the <newline> case. If *line* was not specified, display a screen where the first line of the display shall be (current last line) +1. If there are no lines after the last line in the display, it shall be an error.

38973       ^ If *line* was specified, display a screen where the last line of the display shall contain an unspecified portion of the first line of a display that had an unspecified portion of the specified line on the last line of the display. If this calculation results in a line before the beginning of the edit buffer, display the first screen of the edit buffer.

38977       Otherwise, display a screen where the last line of the display shall contain an unspecified portion of (current first line -1). If this calculation results in a line before the beginning of the edit buffer, it shall be an error.

38980       *Current line:* If *line* and the '^' character were specified:

38981       1. If the first screen was displayed as a result of the command attempting to display lines before the beginning of the edit buffer: if the first screen was already displayed, unchanged; otherwise, set to (current first line -1).

38984       2. Otherwise, set to the last line of the display.

38985       If *line* and the '+' character were specified, set to the first line of the display.

38986       Otherwise, if *line* was specified, set to *line*.

38987       Otherwise, unchanged.

38988       *Current column:* Set to non-<blank>.

38989       **Exit**

38990       *Synopsis:*      ZZ

38991       This command shall be equivalent to the **ex xit** command with no addresses, trailing !, or  
38992       filename (see the **ex xit** command).

38993       **Input Mode Commands in vi**

38994       In text input mode, the current line shall consist of zero or more of the following categories, plus  
38995       the terminating <newline>:

- 38996       1. Characters preceding the text input entry point

38997       Characters in this category shall not be modified during text input mode.

- 38998       2. **autoindent** characters

38999       **autoindent** characters shall be automatically inserted into each line that is created in text  
39000       input mode, either as a result of entering a <newline> or <carriage-return> while in text  
39001       input mode, or as an effect of the command itself; for example, **O** or **o** (see the **ex**  
39002       **autoindent** command), as if entered by the user.

39003       It shall be possible to erase **autoindent** characters with the <control>-D command; it is  
39004       unspecified whether they can be erased by <control>-H, <control>-U, and <control>-W  
39005       characters. Erasing any **autoindent** character turns the glyph into erase-columns and  
39006       deletes the character from the edit buffer, but does not change its representation on the  
39007       screen.

- 39008       3. Text input characters

39009       Text input characters are the characters entered by the user. Erasing any text input  
39010       character turns the glyph into erase-columns and deletes the character from the edit buffer,  
39011       but does not change its representation on the screen.

39012       Each text input character entered by the user (that does not have a special meaning) shall  
39013       be treated as follows:

- 39014       a. The text input character shall be appended to the last character in the edit buffer  
39015           from the first, second, or third categories.

- 39016       b. If there are no erase-columns on the screen, the text input command was the **R**  
39017           command, and characters in the fifth category from the original line follow the  
39018           cursor, the next such character shall be deleted from the edit buffer. If the **slowopen**  
39019           edit option is not set, the corresponding glyph on the screen shall become erase-  
39020           columns.

- 39021       c. If there are erase-columns on the screen, as many columns as they occupy, or as are  
39022           necessary, shall be overwritten to display the text input character. (If only part of a  
39023           multi-column glyph is overwritten, the remainder shall be left on the screen, and  
39024           continue to be treated as erase-columns; it is unspecified whether the remainder of  
39025           the glyph is modified in any way.)

- 39026       d. If additional display line columns are needed to display the text input character:

- 39027           1. If the **slowopen** edit option is set, the text input characters shall be displayed  
39028           on subsequent display line columns, overwriting any characters displayed in

39029 those columns.

2. Otherwise, any characters currently displayed on or after the column on the display line where the text input character is to be displayed shall be pushed ahead the number of display line columns necessary to display the rest of the text input character.

#### 4. Erase-columns

39035 Erase-columns are not logically part of the edit buffer, appearing only on the screen, and  
39036 may be overwritten on the screen by subsequent text input characters. When text input  
39037 mode ends, all erase-columns shall no longer appear on the screen.

39038 Erase-columns are initially the region of text specified by the **c** command (see **Change** (on  
39039 page 1006)); however, erasing **autoindent** or text input characters causes the glyphs of the  
39040 erased characters to be treated as erase-columns.

5. Characters following the text region for the `c` command, or the text input entry point for all other commands

39043 Characters in this category shall not be modified during text input mode, except as  
39044 specified in category 3.b. for the **R** text input command, or as <blank>s deleted when a  
39045 <newline> or <carriage-return> is entered.

It is unspecified whether it is an error to attempt to erase past the beginning of a line that was created by the entry of a <newline> or <carriage-return> during text input mode. If it is not an error, the editor shall behave as if the erasing character was entered immediately after the last text input character entered on the previous line, and all of the non-<newline>s on the current line shall be treated as erase-columns.

When text input mode is entered, or after a text input mode character is entered (except as specified for the special characters below), the cursor shall be positioned as follows:

1. On the first column that displays any part of the first erase-column, if one exists
  2. Otherwise, if the **slowopen** edit option is set, on the first display line column after the last character in the first, second, or third categories, if one exists
  3. Otherwise, the first column that displays any part of the first character in the fifth category, if one exists
  4. Otherwise, the display line column after the last character in the first, second, or third categories, if one exists
  5. Otherwise, on column position 1

39061 The characters that are updated on the screen during text input mode are unspecified, other than  
39062 that the last text input character shall always be updated, and, if the **slowopen** edit option is not  
39063 set, the current cursor character shall always be updated.

39064 The following specifications are for command characters entered during text input mode.

39065       **NUL**

39066       *Synopsis:*     NUL

39067       If the first character of the text input is a NUL, the most recently input text shall be input as if  
39068       entered by the user, and then text input mode shall be exited. The text shall be input literally;  
39069       that is, characters are neither macro or abbreviation expanded, nor are any characters interpreted  
39070       in any special manner. It is unspecified whether implementations shall support more than 256  
39071       bytes of remembered input text.

39072       **<control>-D**

39073       *Synopsis:*     <control>-D

39074       The <control>-D character shall have no special meaning when in text input mode for a line-  
39075       oriented command (see **Command Descriptions in vi** (on page 985)).

39076       This command need not be supported on block-mode terminals.

39077       If the cursor does not follow an **autoindent** character, or an **autoindent** character and a '0' or  
39078       '^' character:

- 39079       1. If the cursor is in column position 1, the <control>-D character shall be discarded and no  
39080       further action taken.
- 39081       2. Otherwise, the <control>-D character shall have no special meaning.

39082       If the last input character was a '0', the cursor shall be moved to column position 1.

39083       Otherwise, if the last input character was a '^', the cursor shall be moved to column position 1.  
39084       In addition, the **autoindent** level for the next input line shall be derived from the same line from  
39085       which the **autoindent** level for the current input line was derived.

39086       Otherwise, the cursor shall be moved back to the column after the previous shiftwidth (see the  
39087       ex **shiftwidth** command) boundary.

39088       All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
39089       cursor position shall become erase-columns as described in **Input Mode Commands in vi** (on  
39090       page 1019).

39091       *Current line:* Unchanged.

39092       *Current column:* Set to 1 if the <control>-D was preceded by a '^' or '0'; otherwise, set to  
39093       (column -1) -((column -2) % **shiftwidth**).

39094       **<control>-H**

39095       *Synopsis:*     <control>-H

39096       If in text input mode for a line-oriented command, and there are no characters to erase, text  
39097       input mode shall be terminated, no further action shall be done for this command, and the  
39098       current line and column shall be unchanged.

39099       If there are characters other than **autoindent** characters that have been input on the current line  
39100       before the cursor, the cursor shall move back one character.

39101       Otherwise, if there are **autoindent** characters on the current line before the cursor, it is  
39102       implementation-defined whether the <control>-H command is an error or if the cursor moves  
39103       back one **autoindent** character.

39104       Otherwise, if the cursor is in column position 1 and there are previous lines that have been input,  
39105       it is implementation-defined whether the <control>-H command is an error or if it is equivalent

39106 to entering <control>-H after the last input character on the previous input line.  
39107 Otherwise, it shall be an error.

39108 All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
39109 cursor position shall become erase-columns as described in **Input Mode Commands in vi** (on  
39110 page 1019).

39111 The current erase character (see *stty*) shall cause an equivalent action to the <control>-H  
39112 command, unless the previously inserted character was a backslash, in which case it shall be as  
39113 if the literal current erase character had been inserted instead of the backslash.

39114 *Current line*: Unchanged, unless previously input lines are erased, in which case it shall be set to  
39115 line -1.

39116 *Current column*: Set to the first column that displays any portion of the character backed up  
39117 over.

39118 <newline>

39119 *Synopsis*: <newline>  
39120 <carriage-return>  
39121 <control>-J  
39122 <control>-M

39123 If input was part of a line-oriented command, text input mode shall be terminated and the  
39124 command shall continue execution with the input provided.

39125 Otherwise, terminate the current line. If there are no characters other than **autoindent** characters  
39126 on the line, all characters on the line shall be discarded. Otherwise, it is unspecified whether the  
39127 **autoindent** characters in the line are modified by entering these characters.

39128 Continue text input mode on a new line appended after the current line. If the **slowopen** edit  
39129 option is set, the lines on the screen below the current line shall not be pushed down, but the  
39130 first of them shall be cleared and shall appear to be overwritten. Otherwise, the lines of the  
39131 screen below the current line shall be pushed down.

39132 If the **autoindent** edit option is set, an appropriate number of **autoindent** characters shall be  
39133 added as a prefix to the line as described by the *ex autoindent* edit option.

39134 All columns after the cursor that are erase-columns (as described in **Input Mode Commands in**  
39135 **vi** (on page 1019)) shall be discarded.

39136 If the **autoindent** edit option is set, all <blank>s immediately following the cursor shall be  
39137 discarded.

39138 All remaining characters after the cursor shall be transferred to the new line, positioned after any  
39139 **autoindent** characters.

39140 *Current line*: Set to current line +1.

39141 *Current column*: Set to the first column that displays any portion of the first character after the  
39142 **autoindent** characters on the new line, if any, or the first column position after the last  
39143 **autoindent** character, if any, or column position 1.

39144       **<control>-T**

39145       *Synopsis:*    **<control>-T**

39146       The **<control>-T** character shall have no special meaning when in text input mode for a line-oriented command (see **Command Descriptions in vi** (on page 985)).

39148       This command need not be supported on block-mode terminals.

39149       Behave as if the user entered the minimum number of **<blank>**s necessary to move the cursor forward to the column position after the next **shiftwidth** (see the **ex shiftwidth** command) boundary.

39152       *Current line:* Unchanged.

39153       *Current column:* Set to *column* + **shiftwidth** – ((*column* – 1) % **shiftwidth**).

39154       **<control>-U**

39155       *Synopsis:*    **<control>-U**

39156       If there are characters other than **autoindent** characters that have been input on the current line before the cursor, the cursor shall move to the first character input after the **autoindent** characters.

39159       Otherwise, if there are **autoindent** characters on the current line before the cursor, it is implementation-defined whether the **<control>-U** command is an error or if the cursor moves to the first column position on the line.

39162       Otherwise, if the cursor is in column position 1 and there are previous lines that have been input, it is implementation-defined whether the **<control>-U** command is an error or if it is equivalent to entering **<control>-U** after the last input character on the previous input line.

39165       Otherwise, it shall be an error.

39166       All of the glyphs on columns between the starting cursor position and (inclusively) the ending cursor position shall become erase-columns as described in **Input Mode Commands in vi** (on page 1019).

39169       The current *kill* character (see **stty**) shall cause an equivalent action to the **<control>-U** command, unless the previously inserted character was a backslash, in which case it shall be as if the literal current *kill* character had been inserted instead of the backslash.

39172       *Current line:* Unchanged, unless previously input lines are erased, in which case it shall be set to line –1.

39174       *Current column:* Set to the first column that displays any portion of the last character backed up over.

39176       **<control>-V**

39177       *Synopsis:*    **<control>-V**

39178              **<control>-Q**

39179       Allow the entry of any subsequent character, other than **<control>-J** or the **<newline>**, as a literal character, removing any special meaning that it may have to the editor in text input mode. If a **<control>-V** or **<control>-Q** is entered before a **<control>-J** or **<newline>**, the **<control>-V** or **<control>-Q** character shall be discarded, and the **<control>-J** or **<newline>** shall behave as described in the **<newline>** command character during input mode.

39184 For purposes of the display only, the editor shall behave as if a '^' character was entered, and  
39185 the cursor shall be positioned as if overwriting the '^' character. When a subsequent character  
39186 is entered, the editor shall behave as if that character was entered instead of the original  
39187 <control>-V or <control>-Q character.

39188 *Current line:* Unchanged.

39189 *Current column:* Unchanged.

39190 <control>-W

39191 *Synopsis:* <control>-W

39192 If there are characters other than **autoindent** characters that have been input on the current line  
39193 before the cursor, the cursor shall move back over the last word preceding the cursor (including  
39194 any <blank>s between the end of the last word and the current cursor); the cursor shall not  
39195 move to before the first character after the end of any **autoindent** characters.

39196 Otherwise, if there are **autoindent** characters on the current line before the cursor, it is  
39197 implementation-defined whether the <control>-W command is an error or if the cursor moves to  
39198 the first column position on the line.

39199 Otherwise, if the cursor is in column position 1 and there are previous lines that have been input,  
39200 it is implementation-defined whether the <control>-W command is an error or if it is equivalent  
39201 to entering <control>-W after the last input character on the previous input line.

39202 Otherwise, it shall be an error.

39203 All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
39204 cursor position shall become erase-columns as described in **Input Mode Commands in vi** (on  
39205 page 1019).

39206 *Current line:* Unchanged, unless previously input lines are erased, in which case it shall be set to  
39207 line -1.

39208 *Current column:* Set to the first column that displays any portion of the last character backed up  
39209 over.

39210 <ESC>

39211 *Synopsis:* <ESC>

39212 If input was part of a line-oriented command:

- 39213 1. If *interrupt* was entered, text input mode shall be terminated and the editor shall return to  
39214 command mode. The terminal shall be alerted.
- 39215 2. If <ESC> was entered, text input mode shall be terminated and the command shall  
39216 continue execution with the input provided.

39217 Otherwise, terminate text input mode and return to command mode.

39218 Any **autoindent** characters entered on newly created lines that have no other non-<newline>s  
39219 shall be deleted.

39220 Any leading **autoindent** and <blank>s on newly created lines shall be rewritten to be the  
39221 minimum number of <blank>s possible.

39222 The screen shall be redisplayed as necessary to match the contents of the edit buffer.

39223 *Current line:* Unchanged.

- 39224        **Current column:**
- 39225        1. If there are text input characters on the current line, the column shall be set to the last  
39226        column where any portion of the last text input character is displayed.
  - 39227        2. Otherwise, if a character is displayed in the current column, unchanged.
  - 39228        3. Otherwise, set to column position 1.

39229        **EXIT STATUS**

39230        The following exit values shall be returned:

- 39231        0 Successful completion.  
39232        >0 An error occurred.

39233        **CONSEQUENCES OF ERRORS**

39234        When any error is encountered and the standard input is not a terminal device file, *vi* shall not  
39235        write the file or return to command or text input mode, and shall terminate with a non-zero exit  
39236        status.

39237        Otherwise, when an unrecoverable error is encountered it shall be equivalent to a SIGHUP  
39238        asynchronous event.

39239        Otherwise, when an error is encountered, the editor shall behave as specified in **Command  
39240        Descriptions in vi** (on page 985).

39241        **APPLICATION USAGE**

39242        None.

39243        **EXAMPLES**

39244        None.

39245        **RATIONALE**

39246        See the RATIONALE for *ex* for more information on *vi*. Major portions of the *vi* utility  
39247        specification point to *ex* to avoid inadvertent divergence. While *ex* and *vi* have historically been  
39248        implemented as a single utility, this is not required by IEEE Std 1003.1-2001.

39249        It is recognized that portions of *vi* would be difficult, if not impossible, to implement  
39250        satisfactorily on a block-mode terminal, or a terminal without any form of cursor addressing,  
39251        thus it is not a mandatory requirement that such features should work on all terminals. It is the  
39252        intention, however, that a *vi* implementation should provide the full set of capabilities on all  
39253        terminals capable of supporting them.

39254        Historically, *vi* exited immediately if the standard input was not a terminal. IEEE Std 1003.1-2001  
39255        permits, but does not require, this behavior. An end-of-file condition is not equivalent to an  
39256        end-of-file character. A common end-of-file character, <control>-D, is historically a *vi* command.

39257        The text in the STDOOUT section reflects the usage of the verb *display* in this section; some  
39258        implementations of *vi* use standard output to write to the terminal, but IEEE Std 1003.1-2001  
39259        does not require that to be the case.

39260        Historically, implementations reverted to open mode if the terminal was incapable of  
39261        supporting full visual mode. IEEE Std 1003.1-2001 requires this behavior. Historically, the open  
39262        mode of *vi* behaved roughly equivalently to the visual mode, with the exception that only a  
39263        single line from the edit buffer (one “buffer line”) was kept current at any time. This line was  
39264        normally displayed on the next-to-last line of a terminal with cursor addressing (and the last line  
39265        performed its normal visual functions for line-oriented commands and messages). In addition,  
39266        some few commands behaved differently in open mode than in visual mode.  
39267        IEEE Std 1003.1-2001 requires conformance to historical practice.

39268 Historically, *ex* and *vi* implementations have expected text to proceed in the usual  
39269 European/Latin order of left to right, top to bottom. There is no requirement in  
39270 IEEE Std 1003.1-2001 that this be the case. The specification was deliberately written using  
39271 words like “before”, “after”, “first”, and “last” in order to permit implementations to support  
39272 the natural text order of the language.

39273 Historically, lines past the end of the edit buffer were marked with single tilde ('~') characters;  
39274 that is, if the one-based display was 20 lines in length, and the last line of the file was on line one,  
39275 then lines 2-20 would contain only a single '~' character.

39276 Historically, the *vi* editor attempted to display only complete lines at the bottom of the screen (it  
39277 did display partial lines at the top of the screen). If a line was too long to fit in its entirety at the  
39278 bottom of the screen, the screen lines where the line would have been displayed were displayed  
39279 as single '@' characters, instead of displaying part of the line. IEEE Std 1003.1-2001 permits, but  
39280 does not require, this behavior. Implementations are encouraged to attempt always to display a  
39281 complete line at the bottom of the screen when doing scrolling or screen positioning by buffer  
39282 lines.

39283 Historically, lines marked with '@' were also used to minimize output to dumb terminals over  
39284 slow lines; that is, changes local to the cursor were updated, but changes to lines on the screen  
39285 that were not close to the cursor were simply marked with an '@' sign instead of being updated  
39286 to match the current text. IEEE Std 1003.1-2001 permits, but does not require this feature because  
39287 it is used ever less frequently as terminals become smarter and connections are faster.

### 39288 Initialization in *ex* and *vi*

39289 Historically, *vi* always had a line in the edit buffer, even if the edit buffer was “empty”. For  
39290 example:

- 39291 1. The *ex* command = executed from visual mode wrote “1” when the buffer was empty.
- 39292 2. Writes from visual mode of an empty edit buffer wrote files of a single character (a  
39293 <newline>), while writes from *ex* mode of an empty edit buffer wrote empty files.
- 39294 3. Put and read commands into an empty edit buffer left an empty line at the top of the edit  
39295 buffer.

39296 For consistency, IEEE Std 1003.1-2001 does not permit any of these behaviors.

39297 Historically, *vi* did not always return the terminal to its original modes; for example, ICRNL was  
39298 modified if it was not originally set. IEEE Std 1003.1-2001 does not permit this behavior.

### 39299 Command Descriptions in *vi*

39300 Motion commands are among the most complicated aspects of *vi* to describe. With some  
39301 exceptions, the text region and buffer type effect of a motion command on a *vi* command are  
39302 described on a case-by-case basis. The descriptions of text regions in IEEE Std 1003.1-2001 are  
39303 not intended to imply direction; that is, an inclusive region from line *n* to line *n+5* is identical to  
39304 a region from line *n+5* to line *n*. This is of more than academic interest—movements to marks  
39305 can be in either direction, and, if the **wrapscan** option is set, so can movements to search points.  
39306 Historically, lines are always stored into buffers in text order; that is, from the start of the edit  
39307 buffer to the end. IEEE Std 1003.1-2001 requires conformance to historical practice.

39308 Historically, command counts were applied to any associated motion, and were multiplicative  
39309 to any supplied motion count. For example, **2cw** is the same as **c2w**, and **2c3w** is the same as  
39310 **c6w**. IEEE Std 1003.1-2001 requires this behavior. Historically, *vi* commands that used bigwords,  
39311 words, paragraphs, and sentences as objects treated groups of empty lines, or lines that  
39312 contained only <blank>s, inconsistently. Some commands treated them as a single entity, while

39313 others treated each line separately. For example, the **w**, **W**, and **B** commands treated groups of  
39314 empty lines as individual words; that is, the command would move the cursor to each new  
39315 empty line. The **e** and **E** commands treated groups of empty lines as a single word; that is, the  
39316 first use would move past the group of lines. The **b** command would just beep at the user, or if  
39317 done from the start of the line as a motion command, fail in unexpected ways. If the lines  
39318 contained only (or ended with) <blank>s, the **w** and **W** commands would just beep at the user,  
39319 the **E** and **e** commands would treat the group as a single word, and the **B** and **b** commands  
39320 would treat the lines as individual words. For consistency and simplicity of specification,  
39321 IEEE Std 1003.1-2001 requires that all *vi* commands treat groups of empty or blank lines as a  
39322 single entity, and that movement through lines ending with <blank>s be consistent with other  
39323 movements.

39324 Historically, *vi* documentation indicated that any number of double quotes were skipped after  
39325 punctuation marks at sentence boundaries; however, implementations only skipped single  
39326 quotes. IEEE Std 1003.1-2001 requires both to be skipped.

39327 Historically, the first and last characters in the edit buffer were word boundaries. This historical  
39328 practice is required by IEEE Std 1003.1-2001.

39329 Historically, *vi* attempted to update the minimum number of columns on the screen possible,  
39330 which could lead to misleading information being displayed. IEEE Std 1003.1-2001 makes no  
39331 requirements other than that the current character being entered is displayed correctly, leaving  
39332 all other decisions in this area up to the implementation.

39333 Historically, lines were arbitrarily folded between columns of any characters that required  
39334 multiple column positions on the screen, with the exception of tabs, which terminated at the  
39335 right-hand margin. IEEE Std 1003.1-2001 permits the former and requires the latter.  
39336 Implementations that do not arbitrarily break lines between columns of characters that occupy  
39337 multiple column positions should not permit the cursor to rest on a column that does not  
39338 contain any part of a character.

39339 The historical *vi* had a problem in that all movements were by buffer lines, not by display or  
39340 screen lines. This is often the right thing to do; for example, single line movements, such as **j** or  
39341 **k**, should work on buffer lines. Commands like **dj**, or **j.**, where . is a change command, only  
39342 make sense for buffer lines. It is not, however, the right thing to do for screen motion or scrolling  
39343 commands like <control>-D, <control>-F, and **H**. If the window is fairly small, using buffer lines  
39344 in these cases can result in completely random motion; for example, **1<control>-D** can result in a  
39345 completely changed screen, without any overlap. This is clearly not what the user wanted. The  
39346 problem is even worse in the case of the **H**, **L**, and **M** commands—as they position the cursor at  
39347 the first non-<blank> of the line, they may all refer to the same location in large lines, and will  
39348 result in no movement at all.

39349 In addition, if the line is larger than the screen, using buffer lines can make it impossible to  
39350 display parts of the line—there are not any commands that do not display the beginning of the  
39351 line in historical *vi*, and if both the beginning and end of the line cannot be on the screen at the  
39352 same time, the user suffers. Finally, the page and half-page scrolling commands historically  
39353 moved to the first non-<blank> in the new line. If the line is approximately the same size as the  
39354 screen, this is inadequate because the cursor before and after a <control>-D command will refer  
39355 to the same location on the screen.

39356 Implementations of *ex* and *vi* exist that do not have these problems because the relevant  
39357 commands (<control>-B, <control>-D, <control>-F, <control>-U, <control>-Y, <control>-E, **H**, **L**,  
39358 and **M**) operate on display (screen) lines, not (edit) buffer lines.

39359 IEEE Std 1003.1-2001 does not permit this behavior by default because the standard developers  
39360 believed that users would find it too confusing. However, historical practice has been relaxed.

For example, *ex* and *vi* historically attempted, albeit sometimes unsuccessfully, to never put part of a line on the last lines of a screen; for example, if a line would not fit in its entirety, no part of the line was displayed, and the screen lines corresponding to the line contained single '@' characters. This behavior is permitted, but not required by IEEE Std 1003.1-2001, so that it is possible for implementations to support long lines in small screens more reasonably without changing the commands to be oriented to the display (instead of oriented to the buffer). IEEE Std 1003.1-2001 also permits implementations to refuse to edit any edit buffer containing a line that will not fit on the screen in its entirety.

The display area (for example, the value of the **window** edit option) has historically been “grown”, or expanded, to display new text when local movements are done in displays where the number of lines displayed is less than the maximum possible. Expansion has historically been the first choice, when the target line is less than the maximum possible expansion value away. Scrolling has historically been the next choice, done when the target line is less than half a display away, and otherwise, the screen was redrawn. There were exceptions, however, in that *ex* commands generally always caused the screen to be redrawn. IEEE Std 1003.1-2001 does not specify a standard behavior because there may be external issues, such as connection speed, the number of characters necessary to redraw as opposed to scroll, or terminal capabilities that implementations will have to accommodate.

The current line in IEEE Std 1003.1-2001 maps one-to-one to a buffer line in the file. The current column does not. There are two different column values that are described by IEEE Std 1003.1-2001. The first is the current column value as set by many of the *vi* commands. This value is remembered for the lifetime of the editor. The second column value is the actual position on the screen where the cursor rests. The two are not always the same. For example, when the cursor is backed by a multi-column character, the actual cursor position on the screen has historically been the last column of the character in command mode, and the first column of the character in input mode.

Commands that set the current line, but that do not set the current cursor value (for example, **j** and **k**) attempt to get as close as possible to the remembered column position, so that the cursor tends to restrict itself to a vertical column as the user moves around in the edit buffer. IEEE Std 1003.1-2001 requires conformance to historical practice, requiring that the display location of the cursor on the display line be adjusted from the current column value as necessary to support this historical behavior.

Historically, only a single line (and for some terminals, a single line minus 1 column) of characters could be entered by the user for the line-oriented commands; that is, ;, !, /, or ?. IEEE Std 1003.1-2001 permits, but does not require, this limitation.

Historically, “soft” errors in *vi* caused the terminal to be alerted, but no error message was displayed. As a general rule, no error message was displayed for errors in command execution in *vi*, when the error resulted from the user attempting an invalid or impossible action, or when a searched-for object was not found. Examples of soft errors included **h** at the left margin, <control>-B or [[ at the beginning of the file, 2G at the end of the file, and so on. In addition, errors such as %, ]], }, N, n, f, F, t, and T failing to find the searched-for object were soft as well. Less consistently, / and ? displayed an error message if the pattern was not found, /, ?, N, and n displayed an error message if no previous regular expression had been specified, and ; did not display an error message if no previous f, F, t, or T command had occurred. Also, behavior in this area might reasonably be based on a runtime evaluation of the speed of a network connection. Finally, some implementations have provided error messages for soft errors in order to assist naive users, based on the value of a verbose edit option. IEEE Std 1003.1-2001 does not list specific errors for which an error message shall be displayed. Implementations should conform to historical practice in the absence of any strong reason to diverge.

39410       **Page Backwards**

39411       The <control>-B and <control>-F commands historically considered it an error to attempt to  
39412       page past the beginning or end of the file, whereas the <control>-D and <control>-U commands  
39413       simply moved to the beginning or end of the file. For consistency, IEEE Std 1003.1-2001 requires  
39414       the latter behavior for all four commands. All four commands still consider it an error if the  
39415       current line is at the beginning (<control>-B, <control>-U) or end (<control>-F, <control>-D) of  
39416       the file. Historically, the <control>-B and <control>-F commands skip two lines in order to  
39417       include overlapping lines when a single command is entered. This makes less sense in the  
39418       presence of a *count*, as there will be, by definition, no overlapping lines. The actual calculation  
39419       used by historical implementations of the vi editor for <control>-B was:

39420        $((\text{current first line}) - \text{count} \times (\text{window edit option})) + 2$

39421       and for <control>-F was:

39422        $((\text{current first line}) + \text{count} \times (\text{window edit option})) - 2$

39423       This calculation does not work well when intermixing commands with and without counts; for  
39424       example, 3<control>-F is not equivalent to entering the <control>-F command three times, and is  
39425       not reversible by entering the <control>-B command three times. For consistency with other vi  
39426       commands that take counts, IEEE Std 1003.1-2001 requires a different calculation.

39427       **Scroll Forward**

39428       The 4BSD and System V implementations of vi differed on the initial value used by the **scroll**  
39429       command. 4BSD used:

39430        $(\text{window edit option}) + 1) / 2$

39431       while System V used the value of the **scroll** edit option. The System V version is specified by  
39432       IEEE Std 1003.1-2001 because the standard developers believed that it was more intuitive and  
39433       permitted the user a method of setting the scroll value initially without also setting the number  
39434       of lines that are displayed.

39435       **Scroll Forward by Line**

39436       Historically, the <control>-E and <control>-Y commands considered it an error if the last and  
39437       first lines, respectively, were already on the screen. IEEE Std 1003.1-2001 requires conformance  
39438       to historical practice. Historically, the <control>-E and <control>-Y commands had no effect in  
39439       open mode. For simplicity and consistency of specification, IEEE Std 1003.1-2001 requires that  
39440       they behave as usual, albeit with a single line screen.

39441       **Clear and Redisplay**

39442       The historical <control>-L command refreshed the screen exactly as it was supposed to be  
39443       currently displayed, replacing any '@' characters for lines that had been deleted but not  
39444       updated on the screen with refreshed '@' characters. The intent of the <control>-L command is  
39445       to refresh when the screen has been accidentally overwritten; for example, by a **write** command  
39446       from another user, or modem noise.

39447

### Redraw Screen

39448

The historical <control>-R command redisplayed only when necessary to update lines that had been deleted but not updated on the screen and that were flagged with '@' characters. There is no requirement that the screen be in any way refreshed if no lines of this form are currently displayed. IEEE Std 1003.1-2001 permits implementations to extend this command to refresh lines on the screen flagged with '@' characters because they are too long to be displayed in the current framework; however, the current line and column need not be modified.

39449

39450

39451

39452

39453

39454

### Search for tagstring

39455

39456

39457

39458

39459

Historically, the first non-<blank> at or after the cursor was the first character, and all subsequent characters that were word characters, up to the end of the line, were included. For example, with the cursor on the leading space or on the '#' character in the text "#bar@", the tag was "#bar". On the character 'b' it was "bar", and on the 'a' it was "ar". IEEE Std 1003.1-2001 requires this behavior.

39460

### Replace Text with Results from Shell Command

39461

39462

39463

39464

39465

Historically, the <, >, and ! commands considered most cursor motions other than line-oriented motions an error; for example, the command >/foo<CR> succeeded, while the command >l failed, even though the text region described by the two commands might be identical. For consistency, all three commands only consider entire lines and not partial lines, and the region is defined as any line that contains a character that was specified by the motion.

39466

### Move to Matching Character

39467

39468

39469

Other matching characters have been left implementation-defined in order to allow extensions such as matching '<' and '>' for searching HTML, or #ifdef, #else, and #endif for searching C source.

39470

### Repeat Substitution

39471

39472

IEEE Std 1003.1-2001 requires that any c and g flags specified to the previous substitute command be ignored; however, the r flag may still apply, if supported by the implementation.

39473

### Return to Previous (Context or Section)

39474

39475

39476

39477

39478

39479

39480

39481

39482

39483

39484

The [[, ]], (, ), {, and } commands are all affected by “section boundaries”, but in some historical implementations not all of the commands recognize the same section boundaries. This is a bug, not a feature, and a unique section-boundary algorithm was not described for each command. One special case that is preserved is that the sentence command moves to the end of the last line of the edit buffer while the other commands go to the beginning, in order to preserve the traditional character cut semantics of the sentence command. Historically, vi section boundaries at the beginning and end of the edit buffer were the first non-<blank> on the first and last lines of the edit buffer if one exists; otherwise, the last character of the first and last lines of the edit buffer if one exists. To increase consistency with other section locations, this has been simplified by IEEE Std 1003.1-2001 to the first character of the first and last lines of the edit buffer, or the first and the last lines of the edit buffer if they are empty.

39485

39486

39487

39488

39489

39490

Sentence boundaries were problematic in the historical vi. They were not only the boundaries as defined for the section and paragraph commands, but they were the first non-<blank> that occurred after those boundaries, as well. Historically, the vi section commands were documented as taking an optional window size as a count preceding the command. This was not implemented in historical versions, so IEEE Std 1003.1-2001 requires that the count repeat the command, for consistency with other vi commands.

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 39491 | <b>Repeat</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 39492 | Historically, mapped commands other than text input commands could not be repeated using the <b>period</b> command. IEEE Std 1003.1-2001 requires conformance to historical practice.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 39494 | The restrictions on the interpretation of special characters (for example, <control>-H) in the repetition of text input mode commands is intended to match historical practice. For example, given the input sequence:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 39497 | iab<control>-H<control>-H<control>-Hdef<escape>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39498 | the user should be informed of an error when the sequence is first entered, but not during a command repetition. The character <control>-T is specifically exempted from this restriction.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 39500 | Historical implementations of <i>vi</i> ignored <control>-T characters that were input in the original command during command repetition. IEEE Std 1003.1-2001 prohibits this behavior.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 39501 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39502 | <b>Find Regular Expression</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 39503 | Historically, commands did not affect the line searched to or from if the motion command was a search (/, ?, N, n) and the final position was the start/end of the line. There were some special cases and <i>vi</i> was not consistent. IEEE Std 1003.1-2001 does not permit this behavior, for consistency. Historical implementations permitted but were unable to handle searches as motion commands that wrapped (that is, due to the edit option <b>wrapscan</b> ) to the original location. IEEE Std 1003.1-2001 requires that this behavior be treated as an error.                                                                                                                                                                                      |
| 39509 | Historically, the syntax "/RE/0" was used to force the command to cut text in line mode. IEEE Std 1003.1-2001 requires conformance to historical practice.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 39510 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39511 | Historically, in open mode, a z specified to a search command redisplayed the current line instead of displaying the current screen with the current line highlighted. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 39512 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39513 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39514 | Historically, trailing z commands were permitted and ignored if entered as part of a search used as a motion command. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 39515 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39516 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39517 | <b>Execute an ex Command</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 39518 | Historically, <i>vi</i> implementations restricted the commands that could be entered on the colon command line (for example, <b>append</b> and <b>change</b> ), and some other commands were known to cause them to fail catastrophically. For consistency, IEEE Std 1003.1-2001 does not permit these restrictions. When executing an <i>ex</i> command by entering :, it is not possible to enter a <newline> as part of the command because it is considered the end of the command. A different approach is to enter <i>ex</i> command mode by using the <i>vi Q</i> command (and later resuming visual mode with the <i>ex vi</i> command). In <i>ex</i> command mode, the single-line limitation does not exist. So, for example, the following is valid: |
| 39519 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39520 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39521 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39522 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39523 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39524 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39525 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39526 | Q                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 39527 | s/break here/break\                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 39528 | here/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 39529 | vi                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 39530 | IEEE Std 1003.1-2001 requires that, if the <i>ex</i> command overwrites any part of the screen that would be erased by a refresh, <i>vi</i> pauses for a character from the user. Historically, this character could be any character; for example, a character input by the user before the message appeared, or even a mapped character. This is probably a bug, but implementations that have tried to be more rigorous by requiring that the user enter a specific character, or that the user enter a character after the message was displayed, have been forced by user indignation back into                                                                                                                                                             |
| 39531 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39532 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39533 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39534 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 39535 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

39536 historical behavior. IEEE Std 1003.1-2001 requires conformance to historical practice.

### 39537 Shift Left (Right)

39538 Refer to the Rationale for the ! and / commands. Historically, the < and > commands sometimes  
39539 moved the cursor to the first non-<blank> (for example if the command was repeated or with \_  
39540 as the motion command), and sometimes left it unchanged. IEEE Std 1003.1-2001 does not  
39541 permit this inconsistency, requiring instead that the cursor always move to the first non-  
39542 <blank>. Historically, the < and > commands did not support buffer arguments, although some  
39543 implementations allow the specification of an optional buffer. This behavior is neither required  
39544 nor disallowed by IEEE Std 1003.1-2001.

### 39545 Execute

39546 Historically, buffers could execute other buffers, and loops, infinite and otherwise, were  
39547 possible. IEEE Std 1003.1-2001 requires conformance to historical practice. The \*buffer syntax of  
39548 ex is not required in vi, because it is not historical practice and has been used in some vi  
39549 implementations to support additional scripting languages.

### 39550 Reverse Case

39551 Historically, the ~ command ignored any associated count, and acted only on the characters in  
39552 the current line. For consistency with other vi commands, IEEE Std 1003.1-2001 requires that an  
39553 associated count act on the next count characters, and that the command move to subsequent  
39554 lines if warranted by count, to make it possible to modify large pieces of text in a reasonably  
39555 efficient manner. There exist vi implementations that optionally require an associated motion  
39556 command for the ~ command. Implementations supporting this functionality are encouraged to  
39557 base it on the **tildedop** edit option and handle the text regions and cursor positioning identically  
39558 to the **yank** command.

### 39559 Append

39560 Historically, counts specified to the A, a, I, and i commands repeated the input of the first line  
39561 count times, and did not repeat the subsequent lines of the input text. IEEE Std 1003.1-2001  
39562 requires that the entire text input be repeated count times.

### 39563 Move Backward to Preceding Word

39564 Historically, vi became confused if word commands were used as motion commands in empty  
39565 files. IEEE Std 1003.1-2001 requires that this be an error. Historical implementations of vi had a  
39566 large number of bugs in the word movement commands, and they varied greatly in behavior in  
39567 the presence of empty lines, “words” made up of a single character, and lines containing only  
39568 <blank>s. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit  
39569 this behavior.

### 39570 Change to End-of-Line

39571 Some historical implementations of the C command did not behave as described by  
39572 IEEE Std 1003.1-2001 when the \$ key was remapped because they were implemented by pushing  
39573 the \$ key onto the input queue and reprocessing it. IEEE Std 1003.1-2001 does not permit this  
39574 behavior. Historically, the C, S, and s commands did not copy replaced text into the numeric  
39575 buffers. For consistency and simplicity of specification, IEEE Std 1003.1-2001 requires that they  
39576 behave like their respective c commands in all respects.

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 39577 | <b>Delete</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 39578 | Historically, lines in open mode that were deleted were scrolled up, and an @ glyph written over the beginning of the line. In the case of terminals that are incapable of the necessary cursor motions, the editor erased the deleted line from the screen. IEEE Std 1003.1-2001 requires conformance to historical practice; that is, if the terminal cannot display the '@' character, the line cannot remain on the screen.                                                                                                                                                           |
| 39583 | <b>Delete to End-of-Line</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 39584 | Some historical implementations of the <b>D</b> command did not behave as described by IEEE Std 1003.1-2001 when the \$ key was remapped because they were implemented by pushing the \$ key onto the input queue and reprocessing it. IEEE Std 1003.1-2001 does not permit this behavior.                                                                                                                                                                                                                                                                                                |
| 39588 | <b>Join</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 39589 | An historical oddity of <i>vi</i> is that the commands <b>J</b> , <b>1J</b> , and <b>2J</b> are all equivalent. IEEE Std 1003.1-2001 requires conformance to historical practice. The <i>vi J</i> command is specified in terms of the <i>ex join</i> command with an <i>ex command count</i> value. The address correction for a <i>count</i> that is past the end of the edit buffer is necessary for historical compatibility for both <i>ex</i> and <i>vi</i> .                                                                                                                       |
| 39594 | <b>Mark Position</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 39595 | Historical practice is that only lowercase letters, plus 'v' and '^', could be used to mark a cursor position. IEEE Std 1003.1-2001 requires conformance to historical practice, but encourages implementations to support other characters as marks as well.                                                                                                                                                                                                                                                                                                                             |
| 39598 | <b>Repeat Regular Expression Find (Forward and Reverse)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 39599 | Historically, the <b>N</b> and <b>n</b> commands could not be used as motion components for the <b>c</b> command. With the exception of the <b>cN</b> command, which worked if the search crossed a line boundary, the text region would be discarded, and the user would not be in text input mode. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.                                                                                                                                                                                 |
| 39603 | <b>Insert Empty Line (Below and Above)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 39604 | Historically, counts to the <b>O</b> and <b>o</b> commands were used as the number of physical lines to open, if the terminal was dumb and the <b>slowopen</b> option was not set. This was intended to minimize traffic over slow connections and repainting for dumb terminals. IEEE Std 1003.1-2001 does not permit this behavior, requiring that a <i>count</i> to the open command behave as for other text input commands. This change to historical practice was made for consistency, and because a superset of the functionality is provided by the <b>slowopen</b> edit option. |
| 39610 | <b>Put from Buffer (Following and Before)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 39611 | Historically, <i>counts</i> to the <b>p</b> and <b>P</b> commands were ignored if the buffer was a line mode buffer, but were (mostly) implemented as described in IEEE Std 1003.1-2001 if the buffer was a character mode buffer. Because implementations exist that do not have this limitation, and because pasting lines multiple times is generally useful, IEEE Std 1003.1-2001 requires that <i>count</i> be supported for all <b>p</b> and <b>P</b> commands.                                                                                                                     |
| 39616 | Historical implementations of <i>vi</i> were widely known to have major problems in the <b>p</b> and <b>P</b> commands, particularly when unusual regions of text were copied into the edit buffer. The standard developers viewed these as bugs, and they are not permitted for consistency and                                                                                                                                                                                                                                                                                          |

39619 simplicity of specification.

39620 Historically, a **P** or **p** command (or an **ex put** command executed from open or visual mode)  
39621 executed in an empty file, left an empty line as the first line of the file. For consistency and  
39622 simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.

### 39623 Replace Character

39624 Historically, the **r** command did not correctly handle the *erase* and *word erase* characters as  
39625 arguments, nor did it handle an associated *count* greater than 1 with a <carriage-return>  
39626 argument, for which it replaced *count* characters with a single <newline>. IEEE Std 1003.1-2001  
39627 does not permit these inconsistencies.

39628 Historically, the **r** command permitted the <control>-V escaping of entered characters, such as  
39629 <ESC> and the <carriage-return>; however, it required two leading <control>-V characters  
39630 instead of one. IEEE Std 1003.1-2001 requires that this be changed for consistency with the other  
39631 text input commands of *vi*.

39632 Historically, it is an error to enter the **r** command if there are less than *count* characters at or after  
39633 the cursor in the line. While a reasonable and unambiguous extension would be to permit the **r**  
39634 command on empty lines, it would require that too large a *count* be adjusted to match the  
39635 number of characters at or after the cursor for consistency, which is sufficiently different from  
39636 historical practice to be avoided. IEEE Std 1003.1-2001 requires conformance to historical  
39637 practice.

### 39638 Replace Characters

39639 Historically, if there were **autoindent** characters in the line on which the **R** command was run,  
39640 and **autoindent** was set, the first <newline> would be properly indented and no characters  
39641 would be replaced by the <newline>. Each additional <newline> would replace *n* characters,  
39642 where *n* was the number of characters that were needed to indent the rest of the line to the  
39643 proper indentation level. This behavior is a bug and is not permitted by IEEE Std 1003.1-2001.

### 39644 Undo

39645 Historical practice for cursor positioning after undoing commands was mixed. In most cases,  
39646 when undoing commands that affected a single line, the cursor was moved to the start of added  
39647 or changed text, or immediately after deleted text. However, if the user had moved from the line  
39648 being changed, the column was either set to the first non-<blank>, returned to the origin of the  
39649 command, or remained unchanged. When undoing commands that affected multiple lines or  
39650 entire lines, the cursor was moved to the first character in the first line restored. As an example  
39651 of how inconsistent this was, a search, followed by an **o** text input command, followed by an  
39652 **undo** would return the cursor to the location where the **o** command was entered, but a **cw**  
39653 command followed by an **o** command followed by an **undo** would return the cursor to the first  
39654 non-<blank> of the line. IEEE Std 1003.1-2001 requires the most useful of these behaviors, and  
39655 discards the least useful, in the interest of consistency and simplicity of specification.

39656

## Yank

39657

Historically, the **yank** command did not move to the end of the motion if the motion was in the forward direction. It moved to the end of the motion if the motion was in the backward direction, except for the **\_** command, or for the **G** and **'** commands when the end of the motion was on the current line. This was further complicated by the fact that for a number of motion commands, the **yank** command moved the cursor but did not update the screen; for example, a subsequent command would move the cursor from the end of the motion, even though the cursor on the screen had not reflected the cursor movement for the **yank** command. IEEE Std 1003.1-2001 requires that all **yank** commands associated with backward motions move the cursor to the end of the motion for consistency, and specifically, to make **'** commands as motions consistent with search patterns as motions.

39658

39659

39660

39661

39662

39663

39664

39665

39666

## Yank Current Line

39667

Some historical implementations of the **Y** command did not behave as described by IEEE Std 1003.1-2001 when the **'** key was remapped because they were implemented by pushing the **'** key onto the input queue and reprocessing it. IEEE Std 1003.1-2001 does not permit this behavior.

39668

## Redraw Window

39669

Historically, the **z** command always redrew the screen. This is permitted but not required by IEEE Std 1003.1-2001, because of the frequent use of the **z** command in macros such as **map n nz**, for screen positioning, instead of its use to change the screen size. The standard developers believed that expanding or scrolling the screen offered a better interface for users. The ability to redraw the screen is preserved if the optional new window size is specified, and in the <control>-L and <control>-R commands.

39670

The semantics of **z^** are confusing at best. Historical practice is that the screen before the screen that ended with the specified line is displayed. IEEE Std 1003.1-2001 requires conformance to historical practice.

39671

Historically, the **z** command would not display a partial line at the top or bottom of the screen. If the partial line would normally have been displayed at the bottom of the screen, the command worked, but the partial line was replaced with '@' characters. If the partial line would normally have been displayed at the top of the screen, the command would fail. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.

39672

Historically, the **z** command with a line specification of 1 ignored the command. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.

39673

Historically, the **z** command did not set the cursor column to the first non-<blank> for the character if the first screen was to be displayed, and was already displayed. For consistency and simplicity of specification, IEEE Std 1003.1-2001 does not permit this behavior.

39674

## Input Mode Commands in vi

39675

Historical implementations of **vi** did not permit the user to erase more than a single line of input, or to use normal erase characters such as *line erase*, *worderase*, and *erase* to erase **autoindent** characters. As there exist implementations of **vi** that do not have these limitations, both behaviors are permitted, but only historical practice is required. In the case of these extensions, **vi** is required to pause at the **autoindent** and previous line boundaries.

39676

Historical implementations of **vi** updated only the portion of the screen where the current cursor character was displayed. For example, consider the **vi** input keystrokes:

39700 iabcd<escape>0C<tab>

39701 Historically, the <tab> would overwrite the characters "abcd" when it was displayed. Other  
39702 implementations replace only the 'a' character with the <tab>, and then push the rest of the  
39703 characters ahead of the cursor. Both implementations have problems. The historical  
39704 implementation is probably visually nicer for the above example; however, for the keystrokes:

39705 iabcd<ESC>0R<tab><ESC>

39706 the historical implementation results in the string "bcd" disappearing and then magically  
39707 reappearing when the <ESC> character is entered. IEEE Std 1003.1-2001 requires the former  
39708 behavior when overwriting erase-columns—that is, overwriting characters that are no longer  
39709 logically part of the edit buffer—and the latter behavior otherwise.

39710 Historical implementations of vi discarded the <control>-D and <control>-T characters when  
39711 they were entered at places where their command functionality was not appropriate.  
39712 IEEE Std 1003.1-2001 requires that the <control>-T functionality always be available, and that  
39713 <control>-D be treated as any other key when not operating on **autoindent** characters.

#### 39714 NUL

39715 Some historical implementations of vi limited the number of characters entered using the NUL  
39716 input character to 256 bytes. IEEE Std 1003.1-2001 permits this limitation; however,  
39717 implementations are encouraged to remove this limit.

#### 39718 <control>-D

39719 See also Rationale for the input mode command <newline>. The hidden assumptions in the  
39720 <control>-D command (and in the vi **autoindent** specification in general) is that <space>s take  
39721 up a single column on the screen and that <tab>s are comprised of an integral number of  
39722 <space>s.

#### 39723 <newline>

39724 Implementations are permitted to rewrite **autoindent** characters in the line when <newline>,  
39725 <carriage-return>, <control>-D, and <control>-T are entered, or when the **shift** commands are  
39726 used, because historical implementations have both done so and found it necessary to do so. For  
39727 example, a <control>-D when the cursor is preceded by a single <tab>, with **tabstop** set to 8, and  
39728 **shiftwidth** set to 3, will result in the <tab> being replaced by several <space>s.

#### 39729 <control>-T

39730 See also the Rationale for the input mode command <newline>. Historically, <control>-T only  
39731 worked if no non-<blank>s had yet been input in the current input line. In addition, the  
39732 characters inserted by <control>-T were treated as **autoindent** characters, and could not be  
39733 erased using normal user erase characters. Because implementations exist that do not have  
39734 these limitations, and as moving to a column boundary is generally useful, IEEE Std 1003.1-2001  
39735 requires that both limitations be removed.

39736 <control>-V

39737 Historically, vi used ^V, regardless of the value of the literal-next character of the terminal.  
39738 IEEE Std 1003.1-2001 requires conformance to historical practice.

39739 The uses described for <control>-V can also be accomplished with <control>-Q, which is useful  
39740 on terminals that use <control>-V for the down-arrow function. However, most historical  
39741 implementations use <control>-Q for the *termios* START character, so the editor will generally  
39742 not receive the <control>-Q unless **stty ixon** mode is set to off. (In addition, some historical  
39743 implementations of vi explicitly set **ixon** mode to on, so it was difficult for the user to set it to  
39744 off.) Any of the command characters described in IEEE Std 1003.1-2001 can be made ineffective  
39745 by their selection as *termios* control characters, using the **stty** utility or other methods described  
39746 in the System Interfaces volume of IEEE Std 1003.1-2001.

39747 <ESC>

39748 Historically, SIGINT alerted the terminal when used to end input mode. This behavior is  
39749 permitted, but not required, by IEEE Std 1003.1-2001.

## 39750 FUTURE DIRECTIONS

39751 None.

## 39752 SEE ALSO

39753 *ed*, *ex*, *stty*

## 39754 CHANGE HISTORY

39755 First released in Issue 2.

### 39756 Issue 5

39757 The FUTURE DIRECTIONS section is added.

### 39758 Issue 6

39759 This utility is marked as part of the User Portability Utilities option.

39760 The APPLICATION USAGE section is added.

39761 The obsolescent SYNOPSIS is removed.

39762 The following new requirements on POSIX implementations derive from alignment with the  
39763 Single UNIX Specification:

- 39764 • The **reindent** command description is added.

39765 The **vi** utility has been extensively rewritten for alignment with the IEEE P1003.2b draft  
39766 standard.

39767 IEEE PASC Interpretations 1003.2 #57, #62, #63, #64, #78, and #188 are applied.

39768 IEEE PASC Interpretation 1003.2 #207 is applied, clarifying the description of the **R** command in  
39769 a manner similar to the descriptions of other text input mode commands such as **i**, **o**, and **O**.

39770 The **-l** option is removed.

**39771 NAME**

39772        wait — await process completion

**39773 SYNOPSIS**

39774        wait [pid...]

**39775 DESCRIPTION**

39776        When an asynchronous list (see Section 2.9.3.1 (on page 50)) is started by the shell, the process ID  
39777        of the last command in each element of the asynchronous list shall become known in the current  
39778        shell execution environment; see Section 2.12 (on page 61).

39779        If the *wait* utility is invoked with no operands, it shall wait until all process IDs known to the  
39780        invoking shell have terminated and exit with a zero exit status.

39781        If one or more *pid* operands are specified that represent known process IDs, the *wait* utility shall  
39782        wait until all of them have terminated. If one or more *pid* operands are specified that represent  
39783        unknown process IDs, *wait* shall treat them as if they were known process IDs that exited with  
39784        exit status 127. The exit status returned by the *wait* utility shall be the exit status of the process  
39785        requested by the last *pid* operand.

39786        The known process IDs are applicable only for invocations of *wait* in the current shell execution  
39787        environment.

**39788 OPTIONS**

39789        None.

**39790 OPERANDS**

39791        The following operand shall be supported:

39792        *pid*        One of the following:

39793        1. The unsigned decimal integer process ID of a command, for which the utility  
39794        is to wait for the termination.

39795        2. A job control job ID (see the Base Definitions volume of IEEE Std 1003.1-2001,  
39796        Section 3.203, Job Control Job ID) that identifies a background process group  
39797        to be waited for. The job control job ID notation is applicable only for  
39798        invocations of *wait* in the current shell execution environment; see Section  
39799        2.12 (on page 61). The exit status of *wait* shall be determined by the last  
39800        command in the pipeline.

39801        **Note:**      The job control job ID type of *pid* is only available on systems supporting  
39802        the User Portability Utilities option.

**39803 STDIN**

39804        Not used.

**39805 INPUT FILES**

39806        None.

**39807 ENVIRONMENT VARIABLES**

39808        The following environment variables shall affect the execution of *wait*:

39809        *LANG*      Provide a default value for the internationalization variables that are unset or null.  
39810        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
39811        Internationalization Variables for the precedence of internationalization variables  
39812        used to determine the values of locale categories.)

39813        *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
39814        internationalization variables.

**39818      *LC\_MESSAGES***  
39819                Determine the locale that should be used to affect the format and contents of  
39820                   diagnostic messages written to standard error.

**39821 XSI NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

39822 ASYNCHRONOUS EVENTS

39823 Default.

39824 STDOUT

39825 Not used.

39826 STDERR

39827 The standard error shall be used only for diagnostic messages.

39828 OUTPUT FILES

39829                  None.

**39830 EXTENDED DESCRIPTION**

39831 None.

## 39832 EXIT STATUS

If one or more operands were specified, all of them have terminated or were not known by the invoking shell, and the status of the last operand specified is known, then the exit status of *wait* shall be the exit status information of the command indicated by the last operand specified. If the process terminated abnormally due to the receipt of a signal, the exit status shall be greater than 128 and shall be distinct from the exit status generated by other signals, but the exact value is unspecified. (See the *kill -l* option.) Otherwise, the *wait* utility shall exit with one of the following values:

39840 0 The *wait* utility was invoked with no operands and all process IDs known by the  
39841 invoking shell have terminated.

39842 1-126 The *wait* utility detected an error.

39843 127 The command identified by the last *pid* operand specified is unknown.

## **39844 CONSEQUENCES OF ERRORS**

39845 Default.

## 39846 APPLICATION USAGE

39847 On most implementations, `wait` is a shell built-in. If it is called in a subshell or separate utility  
39848 execution environment, such as one of the following:

```
39849 (wait)
39850 nohup wait ...
39851 find . -exec wait ... \;
```

it returns immediately because there are no known process IDs to wait for in those environments.

Historical implementations of interactive shells have discarded the exit status of terminated background processes before each shell prompt. Therefore, the status of background processes was usually lost unless it terminated while `wait` was waiting for it. This could be a serious problem when a job that was expected to run for a long time actually terminated quickly with a syntax or initialization error because the exit status returned was usually zero if the requested

39859 process ID was not found. This volume of IEEE Std 1003.1-2001 requires the implementation to  
39860 keep the status of terminated jobs available until the status is requested, so that scripts like:

```
39861 j1&
39862 p1=$!
39863 j2&
39864 wait $p1
39865 echo Job 1 exited with status $?
39866 wait $!
39867 echo Job 2 exited with status $?
```

39868 work without losing status on any of the jobs. The shell is allowed to discard the status of any  
39869 process if it determines that the application cannot get the process ID for that process from the  
39870 shell. It is also required to remember only {CHILD\_MAX} number of processes in this way. Since  
39871 the only way to get the process ID from the shell is by using the '\$!' shell parameter, the shell is  
39872 allowed to discard the status of an asynchronous list if "\$!" was not referenced before another  
39873 asynchronous list was started. (This means that the shell only has to keep the status of the last  
39874 asynchronous list started if the application did not reference "\$!". If the implementation of the  
39875 shell is smart enough to determine that a reference to "\$!" was not saved anywhere that the  
39876 application can retrieve it later, it can use this information to trim the list of saved information.  
39877 Note also that a successful call to *wait* with no operands discards the exit status of all  
39878 asynchronous lists.)

39879 If the exit status of *wait* is greater than 128, there is no way for the application to know if the  
39880 waited-for process exited with that value or was killed by a signal. Since most utilities exit with  
39881 small values, there is seldom any ambiguity. Even in the ambiguous cases, most applications  
39882 just need to know that the asynchronous job failed; it does not matter whether it detected an  
39883 error and failed or was killed and did not complete its job normally.

#### 39884 EXAMPLES

39885 Although the exact value used when a process is terminated by a signal is unspecified, if it is  
39886 known that a signal terminated a process, a script can still reliably determine which signal by  
39887 using *kill* as shown by the following script:

```
39888 sleep 1000&
39889 pid=$!
39890 kill -kill $pid
39891 wait $pid
39892 echo $pid was terminated by a SIG$(kill -l $?) signal.
```

39893 If the following sequence of commands is run in less than 31 seconds:

```
39894 sleep 257 | sleep 31 &
39895 jobs -l %%
```

39896 either of the following commands returns the exit status of the second *sleep* in the pipeline:

```
39897 wait <pid of sleep 31>
39898 wait %%
```

#### 39899 RATIONALE

39900 The description of *wait* does not refer to the *waitpid()* function from the System Interfaces  
39901 volume of IEEE Std 1003.1-2001 because that would needlessly overspecify this interface.  
39902 However, the wording means that *wait* is required to wait for an explicit process when it is given  
39903 an argument so that the status information of other processes is not consumed. Historical  
39904 implementations use the *wait()* function defined in the System Interfaces volume of  
39905 IEEE Std 1003.1-2001 until *wait()* returns the requested process ID or finds that the requested

39906 process does not exist. Because this means that a shell script could not reliably get the status of  
39907 all background children if a second background job was ever started before the first job finished,  
39908 it is recommended that the *wait* utility use a method such as the functionality provided by the  
39909 *waitpid()* function.

39910 The ability to wait for multiple *pid* operands was adopted from the KornShell.

39911 This new functionality was added because it is needed to determine the exit status of any  
39912 asynchronous list accurately. The only compatibility problem that this change creates is for a  
39913 script like

```
39914 while sleep 60 do
39915 job& echo Job started $(date) as $! done
```

39916 which causes the shell to monitor all of the jobs started until the script terminates or runs out of  
39917 memory. This would not be a problem if the loop did not reference "\$!" or if the script would  
39918 occasionally *wait* for jobs it started.

#### 39919 FUTURE DIRECTIONS

39920 None.

#### 39921 SEE ALSO

39922 Chapter 2 (on page 29), *kill*, *sh*, the System Interfaces volume of IEEE Std 1003.1-2001, *wait()*,  
39923 *waitpid()*

#### 39924 CHANGE HISTORY

39925 First released in Issue 2.

**39926 NAME**

39927       wc — word, line, and byte or character count

**39928 SYNOPSIS**

39929       `wc [-c|-m][-lw][file...]`

**39930 DESCRIPTION**

39931       The `wc` utility shall read one or more input files and, by default, write the number of <newline>s, words, and bytes contained in each input file to the standard output.

39933       The utility also shall write a total count for all named files, if more than one input file is specified.

39935       The `wc` utility shall consider a *word* to be a non-zero-length string of characters delimited by white space.

**39937 OPTIONS**

39938       The `wc` utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

39940       The following options shall be supported:

39941       `-c`       Write to the standard output the number of bytes in each input file.

39942       `-l`       Write to the standard output the number of <newline>s in each input file.

39943       `-m`       Write to the standard output the number of characters in each input file.

39944       `-w`       Write to the standard output the number of words in each input file.

39945       When any option is specified, `wc` shall report only the information requested by the specified options.

**39947 OPERANDS**

39948       The following operand shall be supported:

39949       `file`      A pathname of an input file. If no `file` operands are specified, the standard input shall be used.

**39951 STDIN**

39952       The standard input shall be used only if no `file` operands are specified. See the INPUT FILES section.

**39954 INPUT FILES**

39955       The input files may be of any type.

**39956 ENVIRONMENT VARIABLES**

39957       The following environment variables shall affect the execution of `wc`:

39958       `LANG`     Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

39962       `LC_ALL`   If set to a non-empty string value, override the values of all the other internationalization variables.

39964       `LC_CTYPE` Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and which characters are defined as white space characters.

|           |                               |                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 39968     | <b>LC_MESSAGES</b>            | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.                                                                                                                                                                                        |
| 39969     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39970     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39971     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39972 XSI | <b>NLSPATH</b>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                   |
| 39973     | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                                                                                                                                                                                                                         |
| 39974     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                |
| 39975     | <b>STDOUT</b>                 |                                                                                                                                                                                                                                                                                                                                                                         |
| 39976     |                               | By default, the standard output shall contain an entry for each input file of the form:                                                                                                                                                                                                                                                                                 |
| 39977     |                               | "%d %d %d %s\n", <newlines>, <words>, <bytes>, <file>                                                                                                                                                                                                                                                                                                                   |
| 39978     |                               | If the <b>-m</b> option is specified, the number of characters shall replace the <bytes> field in this format.                                                                                                                                                                                                                                                          |
| 39979     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39980     |                               | If any options are specified and the <b>-l</b> option is not specified, the number of <newline>s shall not be written.                                                                                                                                                                                                                                                  |
| 39981     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39982     |                               | If any options are specified and the <b>-w</b> option is not specified, the number of words shall not be written.                                                                                                                                                                                                                                                       |
| 39983     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39984     |                               | If any options are specified and neither <b>-c</b> nor <b>-m</b> is specified, the number of bytes or characters shall not be written.                                                                                                                                                                                                                                  |
| 39985     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39986     |                               | If no input <i>file</i> operands are specified, no name shall be written and no <blank>s preceding the pathname shall be written.                                                                                                                                                                                                                                       |
| 39987     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39988     |                               | If more than one input <i>file</i> operand is specified, an additional line shall be written, of the same format as the other lines, except that the word <b>total</b> (in the POSIX locale) shall be written instead of a pathname and the total of each column shall be written as appropriate. Such an additional line, if any, is written at the end of the output. |
| 39989     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39990     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39991     |                               |                                                                                                                                                                                                                                                                                                                                                                         |
| 39992     | <b>STDERR</b>                 |                                                                                                                                                                                                                                                                                                                                                                         |
| 39993     |                               | The standard error shall be used only for diagnostic messages.                                                                                                                                                                                                                                                                                                          |
| 39994     | <b>OUTPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                         |
| 39995     |                               | None.                                                                                                                                                                                                                                                                                                                                                                   |
| 39996     | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                                                                                                                                                                                                                         |
| 39997     |                               | None.                                                                                                                                                                                                                                                                                                                                                                   |
| 39998     | <b>EXIT STATUS</b>            |                                                                                                                                                                                                                                                                                                                                                                         |
| 39999     |                               | The following exit values shall be returned:                                                                                                                                                                                                                                                                                                                            |
| 40000     |                               | 0 Successful completion.                                                                                                                                                                                                                                                                                                                                                |
| 40001     |                               | >0 An error occurred.                                                                                                                                                                                                                                                                                                                                                   |
| 40002     | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                                                                                                                                                                                                                         |
| 40003     |                               | Default.                                                                                                                                                                                                                                                                                                                                                                |

## 40004 APPLICATION USAGE

40005       The **-m** option is not a switch, but an option at the same level as **-c**. Thus, to produce the full  
40006       default output with character counts instead of bytes, the command required is:

40007       `wc -mlw`

## 40008 EXAMPLES

40009       None.

## 40010 RATIONALE

40011       The output file format pseudo-*printf()* string differs from the System V version of *wc*:

40012       `"%7d%7d%7d %s\n"`

40013       which produces possibly ambiguous and unparsable results for very large files, as it assumes no  
40014       number shall exceed six digits.

40015       Some historical implementations use only <space>, <tab>, and <newline> as word separators.  
40016       The equivalent of the ISO C standard *isspace()* function is more appropriate.

40017       The **-c** option stands for “character” count, even though it counts bytes. This stems from the  
40018       sometimes erroneous historical view that bytes and characters are the same size. Due to  
40019       international requirements, the **-m** option (reminiscent of “multi-byte”) was added to obtain  
40020       actual character counts.

40021       Early proposals only specified the results when input files were text files. The current  
40022       specification more closely matches historical practice. (Bytes, words, and <newline>s are  
40023       counted separately and the results are written when an end-of-file is detected.)

40024       Historical implementations of the *wc* utility only accepted one argument to specify the options  
40025       **-c**, **-l**, and **-w**. Some of them also had multiple occurrences of an option cause the  
40026       corresponding count to be written multiple times and had the order of specification of the  
40027       options affect the order of the fields on output, but did not document either of these. Because  
40028       common usage either specifies no options or only one option, and because none of this was  
40029       documented, the changes required by this volume of IEEE Std 1003.1-2001 should not break  
40030       many historical applications (and do not break any historical conforming applications).

## 40031 FUTURE DIRECTIONS

40032       None.

## 40033 SEE ALSO

40034       *cksum*

## 40035 CHANGE HISTORY

40036       First released in Issue 2.

40037 **NAME**

40038        what — identify SCCS files (**DEVELOPMENT**)

40039 **SYNOPSIS**

40040 XSI        **what [-s] file...**

40041

40042 **DESCRIPTION**

40043        The *what* utility shall search the given files for all occurrences of the pattern that *get* (see *get*)  
 40044        substitutes for the %Z% keyword ("@(#)") and shall write to standard output what follows  
 40045        until the first occurrence of one of the following:

40046        " > newline \ NUL

40047 **OPTIONS**

40048        The *what* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section  
 40049        12.2, Utility Syntax Guidelines.

40050        The following option shall be supported:

40051        **-s**        Quit after finding the first occurrence of the pattern in each file.

40052 **OPERANDS**

40053        The following operands shall be supported:

40054        *file*        A pathname of a file to search.

40055 **STDIN**

40056        Not used.

40057 **INPUT FILES**

40058        The input files shall be of any file type.

40059 **ENVIRONMENT VARIABLES**

40060        The following environment variables shall affect the execution of *what*:

40061        **LANG**        Provide a default value for the internationalization variables that are unset or null.  
 40062        (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
 40063        Internationalization Variables for the precedence of internationalization variables  
 40064        used to determine the values of locale categories.)

40065        **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
 40066        internationalization variables.

40067        **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
 40068        characters (for example, single-byte as opposed to multi-byte characters in  
 40069        arguments and input files).

40070        **LC\_MESSAGES**

40071        Determine the locale that should be used to affect the format and contents of  
 40072        diagnostic messages written to standard error.

40073        **NLSPATH**     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

40074 **ASYNCHRONOUS EVENTS**

40075        Default.

40076 **STDOUT**

40077        The standard output shall consist of the following for each *file* operand:

40078        "%s:\n\t%s\n", <pathname>, <identification string>

**40079 STDERR**

40080 The standard error shall be used only for diagnostic messages.

**40081 OUTPUT FILES**

40082 None.

**40083 EXTENDED DESCRIPTION**

40084 None.

**40085 EXIT STATUS**

40086 The following exit values shall be returned:

40087 0 Any matches were found.

40088 1 Otherwise.

**40089 CONSEQUENCES OF ERRORS**

40090 Default.

**40091 APPLICATION USAGE**

40092 The *what* utility is intended to be used in conjunction with the SCCS command *get*, which  
40093 automatically inserts identifying information, but it can also be used where the information is  
40094 inserted by any other means.

40095 When the string "@(#)" is included in a library routine in a shared library, it might not be found  
40096 in an **a.out** file using that library routine.

**40097 EXAMPLES**

40098 If the C-language program in file **f.c** contains:

40099 char ident[ ] = "@(#)identification information";

40100 and **f.c** is compiled to yield **f.o** and **a.out**, then the command:

40101 what f.c f.o a.out

40102 writes:

40103 f.c:  
40104 identification information

40105 ...

40106 f.o:  
40107 identification information  
40108 ...  
40109 a.out:  
40110 identification information  
40111 ...

**40112 RATIONALE**

40113 None.

**40114 FUTURE DIRECTIONS**

40115 None.

**40116 SEE ALSO**

40117 *get*

**40118 CHANGE HISTORY**

40119 First released in Issue 2.

## 40120 NAME

40121 who — display who is on the system

## 40122 SYNOPSIS

40123 UP who [-mTu]

40124 XSI who [-mu]-s[-bHlprt][file]

40125 who [-mTu][-abdHlprt][file]

40126 who -q [file]

40127 who am i

40128 who am I

40129

## 40130 DESCRIPTION

40131 The *who* utility shall list various pieces of information about accessible users. The domain of accessibility is implementation-defined.

40133 XSI Based on the options given, *who* can also list the user's name, terminal line, login time, elapsed time since activity occurred on the line, and the process ID of the command interpreter for each current system user.

## 40136 OPTIONS

40137 The *who* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

40139 The following options shall be supported. The metavariables, such as <line>, refer to fields described in the STDOUT section.

40141 XSI **-a** Process the implementation-defined database or named file with the **-b**, **-d**, **-l**, **-p**, **-r**, **-t**, **-T** and **-u** options turned on.

40143 XSI **-b** Write the time and date of the last reboot.

40144 XSI **-d** Write a list of all processes that have expired and not been respawned by the *init* system process. The <exit> field shall appear for dead processes and contain the termination and exit values of the dead process. This can be useful in determining why a process terminated.

40148 XSI **-H** Write column headings above the regular output.

40149 XSI **-l** (The letter ell.) List only those lines on which the system is waiting for someone to login. The <name> field shall be LOGIN in such cases. Other fields shall be the same as for user entries except that the <state> field does not exist.

40152 **-m** Output only information about the current terminal.

40153 XSI **-p** List any other process that is currently active and has been previously spawned by *init*.

40155 XSI **-q** (Quick.) List only the names and the number of users currently logged on. When this option is used, all other options shall be ignored.

40157 XSI **-r** Write the current run-level of the *init* process.

40158 XSI **-s** List only the <name>, <line>, and <time> fields. This is the default case.

40159 XSI **-t** Indicate the last change to the system clock.

|           |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40160     | <b>-T</b> | Show the state of each terminal, as described in the STDOUT section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 40161     | <b>-u</b> | Write “idle time” for each displayed user in addition to any other information. The idle time is the time since any activity occurred on the user’s terminal. The method of determining this is unspecified. This option shall list only those users who are currently logged in. The <i>&lt;name&gt;</i> is the user’s login name. The <i>&lt;line&gt;</i> is the name of the line as found in the directory /dev. The <i>&lt;time&gt;</i> is the time that the user logged in. The <i>&lt;activity&gt;</i> is the number of hours and minutes since activity last occurred on that particular line. A dot indicates that the terminal has seen activity in the last minute and is therefore “current”. If more than twenty-four hours have elapsed or the line has not been used since boot time, the entry shall be marked <i>&lt;old&gt;</i> . This field is useful when trying to determine whether a person is working at the terminal or not. The <i>&lt;pid&gt;</i> is the process ID of the user’s login process. |
| 40162     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40163 XSI |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40164     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40165     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40166     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40167     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40168     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40169     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40170     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40171     |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## 40172 OPERANDS

|           |                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40173 XSI | The following operands shall be supported:                                                                                                                                                      |
| 40174     | <b>am i, am I</b> In the POSIX locale, limit the output to describing the invoking user, equivalent to the <b>-m</b> option. The <b>am</b> and <b>i</b> or <b>I</b> must be separate arguments. |
| 40175     |                                                                                                                                                                                                 |
| 40176     | <b>file</b> Specify a pathname of a file to substitute for the implementation-defined database of logged-on users that <i>who</i> uses by default.                                              |
| 40177     |                                                                                                                                                                                                 |

## 40178 STDIN

40179 Not used.

## 40180 INPUT FILES

40181 None.

## 40182 ENVIRONMENT VARIABLES

|           |                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40183     | The following environment variables shall affect the execution of <i>who</i> :                                                                                                                                                                                                                                    |
| 40184     | <b>LANG</b> Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.) |
| 40185     |                                                                                                                                                                                                                                                                                                                   |
| 40186     |                                                                                                                                                                                                                                                                                                                   |
| 40187     |                                                                                                                                                                                                                                                                                                                   |
| 40188     | <b>LC_ALL</b> If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                            |
| 40189     |                                                                                                                                                                                                                                                                                                                   |
| 40190     | <b>LC_CTYPE</b> Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                         |
| 40191     |                                                                                                                                                                                                                                                                                                                   |
| 40192     |                                                                                                                                                                                                                                                                                                                   |
| 40193     | <b>LC_MESSAGES</b>                                                                                                                                                                                                                                                                                                |
| 40194     | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                      |
| 40195     |                                                                                                                                                                                                                                                                                                                   |
| 40196     | <b>LC_TIME</b> Determine the locale used for the format and contents of the date and time strings.                                                                                                                                                                                                                |
| 40197 XSI | <b>NLSPATH</b> Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                                                                                                                                                                                              |
| 40198     | <b>TZ</b> Determine the timezone used when writing date and time information. If <b>TZ</b> is unset or null, an unspecified default timezone shall be used.                                                                                                                                                       |
| 40199     |                                                                                                                                                                                                                                                                                                                   |

## 40200 ASYNCHRONOUS EVENTS

40201 Default.

**40202 STDOUT**

40203 The *who* utility shall write its default format to the standard output in an implementation-defined format, subject only to the requirement of containing the information described above.

40205 XSI OF XSI-conformant systems shall write the default information to the standard output in the  
40206 following general format:

40207 `<name>[<state>]<line><time>[<activity>][<pid>][<comment>][<exit>]`

40208 The following format shall be used for the **-T** option:

40209 `"%s %c %s %s\n" <name>, <terminal state>, <terminal name>,`  
40210 `<time of login>`

40211 where *<terminal state>* is one of the following characters:

- 40212 + The terminal allows write access to other users.
- 40213 - The terminal denies write access to other users.
- 40214 ? The terminal write-access state cannot be determined.

40215 In the POSIX locale, the *<time of login>* shall be equivalent in format to the output of:

40216 `date + "%b %e %H:%M"`

40217 If the **-u** option is used with **-T**, the idle time shall be added to the end of the previous format in  
40218 an unspecified format.

**40219 STDERR**

40220 The standard error shall be used only for diagnostic messages.

**40221 OUTPUT FILES**

40222 None.

**40223 EXTENDED DESCRIPTION**

40224 None.

**40225 EXIT STATUS**

40226 The following exit values shall be returned:

- 40227 0 Successful completion.
- 40228 >0 An error occurred.

**40229 CONSEQUENCES OF ERRORS**

40230 Default.

**40231 APPLICATION USAGE**

40232 The name *init* used for the system process is the most commonly used on historical systems, but  
40233 it may vary.

40234 The “domain of accessibility” referred to is a broad concept that permits interpretation either on  
40235 a very secure basis or even to allow a network-wide implementation like the historical *rwho*.

**40236 EXAMPLES**

40237 None.

**40238 RATIONALE**

40239 Due to differences between historical implementations, the base options provided were a  
40240 compromise to allow users to work with those functions. The standard developers also  
40241 considered removing all the options, but felt that these options offered users valuable  
40242 functionality. Additional options to match historical systems are available on XSI-conformant

40243 systems.

40244 It is recognized that the *who* command may be of limited usefulness, especially in a multi-level  
40245 secure environment. The standard developers considered, however, that having some standard  
40246 method of determining the “accessibility” of other users would aid user portability.

40247 No format was specified for the default *who* output for systems not supporting the XSI  
40248 Extension. In such a user-oriented command, designed only for human use, this was not  
40249 considered to be a deficiency.

40250 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, and *write* require  
40251 that they use the same format.

40252 It is acceptable for an implementation to produce no output for an invocation of *who mil*.

40253 **FUTURE DIRECTIONS**

40254 None.

40255 **SEE ALSO**

40256 *mesg*

40257 **CHANGE HISTORY**

40258 First released in Issue 2.

40259 **Issue 6**

40260 This utility is marked as part of the User Portability Utilities option.

40261 The *TZ* entry is added to the ENVIRONMENT VARIABLES section.

## 40262 NAME

40263 write — write to another user

## 40264 SYNOPSIS

40265 UP write user\_name [terminal]

40266

## 40267 DESCRIPTION

40268 The *write* utility shall read lines from the user's standard input and write them to the terminal of  
40269 another user. When first invoked, it shall write the message:

40270 **Message from sender-login-id (sending-terminal) [date]...**

40271 to *user\_name*. When it has successfully completed the connection, the sender's terminal shall be  
40272 alerted twice to indicate that what the sender is typing is being written to the recipient's  
40273 terminal.

40274 If the recipient wants to reply, this can be accomplished by typing:

40275 write sender-login-id [sending-terminal]

40276 upon receipt of the initial message. Whenever a line of input as delimited by an NL, EOF, or EOL  
40277 special character (see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General  
40278 Terminal Interface) is accumulated while in canonical input mode, the accumulated data shall be  
40279 written on the other user's terminal. Characters shall be processed as follows:

- 40280 • Typing <alert> shall write the alert character to the recipient's terminal.
- 40281 • Typing the erase and kill characters shall affect the sender's terminal in the manner described  
40282 by the **termios** interface in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11,  
40283 General Terminal Interface.
- 40284 • Typing the interrupt or end-of-file characters shall cause *write* to write an appropriate  
40285 message ("EOT\n" in the POSIX locale) to the recipient's terminal and exit.
- 40286 • Typing characters from *LC\_CTYPE* classifications **print** or **space** shall cause those characters  
40287 to be sent to the recipient's terminal.
- 40288 • When and only when the *stty iexten* local mode is enabled, the existence and processing of  
40289 additional special control characters and multi-byte or single-byte functions is  
40290 implementation-defined.
- 40291 • Typing other non-printable characters shall cause implementation-defined sequences of  
40292 printable characters to be written to the recipient's terminal.

40293 To write to a user who is logged in more than once, the *terminal* argument can be used to indicate  
40294 which terminal to write to; otherwise, the recipient's terminal is selected in an implementation-  
40295 defined manner and an informational message is written to the sender's standard output,  
40296 indicating which terminal was chosen.

40297 Permission to be a recipient of a *write* message can be denied or granted by use of the *mesg*  
40298 utility. However, a user's privilege may further constrain the domain of accessibility of other  
40299 users' terminals. The *write* utility shall fail when the user lacks the appropriate privileges to  
40300 perform the requested action.

## 40301 OPTIONS

40302 None.

**40303 OPERANDS**

40304 The following operands shall be supported:

40305 *user\_name* Login name of the person to whom the message shall be written. The application  
40306 shall ensure that this operand is of the form returned by the *who* utility.

40307 *terminal* Terminal identification in the same format provided by the *who* utility.

**40308 STDIN**

40309 Lines to be copied to the recipient's terminal are read from standard input.

**40310 INPUT FILES**

40311 None.

**40312 ENVIRONMENT VARIABLES**

40313 The following environment variables shall affect the execution of *write*:

40314 *LANG* Provide a default value for the internationalization variables that are unset or null.  
40315 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
40316 Internationalization Variables for the precedence of internationalization variables  
40317 used to determine the values of locale categories.)

40318 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
40319 internationalization variables.

40320 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
40321 characters (for example, single-byte as opposed to multi-byte characters in  
40322 arguments and input files). If the recipient's locale does not use an *LC\_CTYPE*  
40323 equivalent to the sender's, the results are undefined.

**40324 *LC\_MESSAGES***

40325 Determine the locale that should be used to affect the format and contents of  
40326 diagnostic messages written to standard error and informative messages written to  
40327 standard output.

40328 XSI ***NLSPATH*** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**40329 ASYNCHRONOUS EVENTS**

40330 If an interrupt signal is received, *write* shall write an appropriate message on the recipient's  
40331 terminal and exit with a status of zero. It shall take the standard action for all other signals.

**40332 STDOUT**

40333 An informational message shall be written to standard output if a recipient is logged in more  
40334 than once.

**40335 STDERR**

40336 The standard error shall be used only for diagnostic messages.

**40337 OUTPUT FILES**

40338 The recipient's terminal is used for output.

**40339 EXTENDED DESCRIPTION**

40340 None.

**40341 EXIT STATUS**

40342 The following exit values shall be returned:

40343 0 Successful completion.

40344 >0 The addressed user is not logged on or the addressed user denies permission.

**40345 CONSEQUENCES OF ERRORS**

40346 Default.

**40347 APPLICATION USAGE**

40348 The *talk* utility is considered by some users to be a more usable utility on full-screen terminals.

**40349 EXAMPLES**

40350 None.

**40351 RATIONALE**

40352 The *write* utility was included in this volume of IEEE Std 1003.1-2001 since it can be  
40353 implemented on all terminal types. The standard developers considered the *talk* utility, which  
40354 cannot be implemented on certain terminals, to be a “better” communications interface. Both of  
40355 these programs are in widespread use on historical implementations. Therefore, the standard  
40356 developers decided that both utilities should be specified.

40357 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
40358 require that they all use or accept the same format.

**40359 FUTURE DIRECTIONS**

40360 None.

**40361 SEE ALSO**

40362 *mesg*, *talk*, *who*, the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 11, General  
40363 Terminal Interface

**40364 CHANGE HISTORY**

40365 First released in Issue 2.

**40366 Issue 5**

40367 The FUTURE DIRECTIONS section is added.

**40368 Issue 6**

40369 This utility is marked as part of the User Portability Utilities option.

40370 The normative text is reworded to avoid use of the term “must” for application requirements.

## 40371 NAME

40372        xargs — construct argument lists and invoke utility

## 40373 SYNOPSIS

40374 XSI        xargs [-t][-p][-E eofstr][-I replstr][-L number][-n number [-x]]  
40375            [-s size][utility [argument...]]

## 40376 DESCRIPTION

40377        The *xargs* utility shall construct a command line consisting of the *utility* and *argument* operands specified followed by as many arguments read in sequence from standard input as fit in length and number constraints specified by the options. The *xargs* utility shall then invoke the constructed command line and wait for its completion. This sequence shall be repeated until one of the following occurs:

- 40382        • An end-of-file condition is detected on standard input.
- 40383        • The logical end-of-file string (see the **-E eofstr** option) is found on standard input after double-quote processing, apostrophe processing, and backslash escape processing (see next paragraph).
- 40386        • An invocation of a constructed command line returns an exit status of 255.

40387        The application shall ensure that arguments in the standard input are separated by unquoted <blank>s, unescaped <blank>s, or <newline>s. A string of zero or more non-double-quote ('"') characters and non-<newline>s can be quoted by enclosing them in double-quotes. A string of zero or more non-apostrophe ('') characters and non-<newline>s can be quoted by enclosing them in apostrophes. Any unquoted character can be escaped by preceding it with a backslash. The utility named by *utility* shall be executed one or more times until the end-of-file is reached or the logical end-of file string is found. The results are unspecified if the utility named by *utility* attempts to read from its standard input.

40395        The generated command line length shall be the sum of the size in bytes of the utility name and each argument treated as strings, including a null byte terminator for each of these strings. The *xargs* utility shall limit the command line length such that when the command line is invoked, the combined argument and environment lists (see the *exec* family of functions in the System Interfaces volume of IEEE Std 1003.1-2001) shall not exceed {ARG\_MAX}-2 048 bytes. Within this constraint, if neither the **-n** nor the **-s** option is specified, the default command line length shall be at least {LINE\_MAX}.

## 40402 OPTIONS

40403        The *xargs* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

40405        The following options shall be supported:

40406        **-E eofstr**        Use *eofstr* as the logical end-of-file string. If **-E** is not specified, it is unspecified whether the logical end-of-file string is the underscore character ('\_') or the end-of-file string capability is disabled. When *eofstr* is the null string, the logical end-of-file string capability shall be disabled and underscore characters shall be taken literally.

40411 XSI        **-I replstr**        Insert mode: *utility* is executed for each line from standard input, taking the entire line as a single argument, inserting it in *arguments* for each occurrence of *replstr*. A maximum of five arguments in *arguments* can each contain one or more instances of *replstr*. Any <blank>s at the beginning of each line shall be ignored. Constructed arguments cannot grow larger than 255 bytes. Option **-x** shall be forced on.

|           |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40417 XSI | <b>-L</b> <i>number</i> | The <i>utility</i> shall be executed for each non-empty <i>number</i> lines of arguments from standard input. The last invocation of <i>utility</i> shall be with fewer lines of arguments if fewer than <i>number</i> remain. A line is considered to end with the first <newline> unless the last character of the line is a <blank>; a trailing <blank> signals continuation to the next non-empty line, inclusive. The <b>-L</b> and <b>-n</b> options are mutually-exclusive; the last one specified shall take effect. |
| 40423     | <b>-n</b> <i>number</i> | Invoke <i>utility</i> using as many standard input arguments as possible, up to <i>number</i> (a positive decimal integer) arguments maximum. Fewer arguments shall be used if:                                                                                                                                                                                                                                                                                                                                              |
| 40425     |                         | <ul style="list-style-type: none"> <li>• The command line length accumulated exceeds the size specified by the <b>-s</b> option (or {LINE_MAX} if there is no <b>-s</b> option).</li> </ul>                                                                                                                                                                                                                                                                                                                                  |
| 40426     |                         | <ul style="list-style-type: none"> <li>• The last iteration has fewer than <i>number</i>, but not zero, operands remaining.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                       |
| 40428     | <b>-p</b>               | Prompt mode: the user is asked whether to execute <i>utility</i> at each invocation. Trace mode ( <b>-t</b> ) is turned on to write the command instance to be executed, followed by a prompt to standard error. An affirmative response read from /dev/tty shall execute the command; otherwise, that particular invocation of <i>utility</i> shall be skipped.                                                                                                                                                             |
| 40433     | <b>-s</b> <i>size</i>   | Invoke <i>utility</i> using as many standard input arguments as possible yielding a command line length less than <i>size</i> (a positive decimal integer) bytes. Fewer arguments shall be used if:                                                                                                                                                                                                                                                                                                                          |
| 40436     |                         | <ul style="list-style-type: none"> <li>• The total number of arguments exceeds that specified by the <b>-n</b> option.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                            |
| 40437 XSI |                         | <ul style="list-style-type: none"> <li>• The total number of lines exceeds that specified by the <b>-L</b> option.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                |
| 40438     |                         | <ul style="list-style-type: none"> <li>• End-of-file is encountered on standard input before <i>size</i> bytes are accumulated.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                   |
| 40439     |                         | Values of <i>size</i> up to at least {LINE_MAX} bytes shall be supported, provided that the constraints specified in the DESCRIPTION are met. It shall not be considered an error if a value larger than that supported by the implementation or exceeding the constraints specified in the DESCRIPTION is given; xargs shall use the largest value it supports within the constraints.                                                                                                                                      |
| 40444     | <b>-t</b>               | Enable trace mode. Each generated command line shall be written to standard error just prior to invocation.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 40446     | <b>-x</b>               | Terminate if a command line containing <i>number</i> arguments (see the <b>-n</b> option above) or <i>number</i> lines (see the <b>-L</b> option above) will not fit in the implied or specified size (see the <b>-s</b> option above).                                                                                                                                                                                                                                                                                      |
| 40447 XSI |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 40448     |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

**40449 OPERANDS**

40450 The following operands shall be supported:

|       |                 |                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40451 | <i>utility</i>  | The name of the utility to be invoked, found by search path using the <i>PATH</i> environment variable, described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables. If <i>utility</i> is omitted, the default shall be the <i>echo</i> utility. If the <i>utility</i> operand names any of the special built-in utilities in Section 2.14 (on page 64), the results are undefined. |
| 40456 | <i>argument</i> | An initial option or operand for the invocation of <i>utility</i> .                                                                                                                                                                                                                                                                                                                                                        |

**40457 STDIN**

40458 The standard input shall be a text file. The results are unspecified if an end-of-file condition is detected immediately following an escaped <newline>.

**40460 INPUT FILES**

40461 The file `/dev/tty` shall be used to read responses required by the `-p` option.

**40462 ENVIRONMENT VARIABLES**

40463 The following environment variables shall affect the execution of *xargs*:

40464 *LANG* Provide a default value for the internationalization variables that are unset or null.  
(See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

40468 *LC\_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

*LC\_COLLATE*

40471 Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the extended regular expression defined for the `yesexpr` locale keyword in the *LC\_MESSAGES* category.

40474 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes used in the extended regular expression defined for the `yesexpr` locale keyword in the *LC\_MESSAGES* category.

*LC\_MESSAGES*

40480 Determine the locale for the processing of affirmative responses and that should be used to affect the format and contents of diagnostic messages written to standard error.

40483 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

40484 PATH Determine the location of *utility*, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 8, Environment Variables.

**40486 ASYNCHRONOUS EVENTS**

40487 Default.

**40488 STDOUT**

40489 Not used.

**40490 STDERR**

40491 The standard error shall be used for diagnostic messages and the `-t` and `-p` options. If the `-t` option is specified, the *utility* and its constructed argument list shall be written to standard error, as it will be invoked, prior to invocation. If `-p` is specified, a prompt of the following format shall be written (in the POSIX locale):

40495 " ? . . . "

40496 at the end of the line of the output from `-t`.

**40497 OUTPUT FILES**

40498 None.

**40499 EXTENDED DESCRIPTION**

40500 None.

**40501 EXIT STATUS**

40502 The following exit values shall be returned:

- 40503 0 All invocations of *utility* returned exit status zero.
- 40504 1-125 A command line meeting the specified requirements could not be assembled, one or  
40505 more of the invocations of *utility* returned a non-zero exit status, or some other error  
40506 occurred.
- 40507 126 The utility specified by *utility* was found but could not be invoked.
- 40508 127 The utility specified by *utility* could not be found.

**40509 CONSEQUENCES OF ERRORS**

40510 If a command line meeting the specified requirements cannot be assembled, the utility cannot be  
40511 invoked, an invocation of the utility is terminated by a signal, or an invocation of the utility exits  
40512 with exit status 255, the *xargs* utility shall write a diagnostic message and exit without  
40513 processing any remaining input.

**40514 APPLICATION USAGE**

40515 The 255 exit status allows a utility being used by *xargs* to tell *xargs* to terminate if it knows no  
40516 further invocations using the current data stream will succeed. Thus, *utility* should explicitly *exit*  
40517 with an appropriate value to avoid accidentally returning with 255.

40518 Note that input is parsed as lines; <blank>s separate arguments. If *xargs* is used to bundle output  
40519 of commands like *find dir -print* or *ls* into commands to be executed, unexpected results are  
40520 likely if any filenames contain any <blank>s or <newline>s. This can be fixed by using *find* to  
40521 call a script that converts each file found into a quoted string that is then piped to *xargs*. Note  
40522 that the quoting rules used by *xargs* are not the same as in the shell. They were not made  
40523 consistent here because existing applications depend on the current rules and the shell syntax is  
40524 not fully compatible with it. An easy rule that can be used to transform any string into a quoted  
40525 form that *xargs* interprets correctly is to precede each character in the string with a backslash.

40526 On implementations with a large value for {ARG\_MAX}, *xargs* may produce command lines  
40527 longer than {LINE\_MAX}. For invocation of utilities, this is not a problem. If *xargs* is being used  
40528 to create a text file, users should explicitly set the maximum command line length with the *-s*  
40529 option.

40530 The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if  
40531 an error occurs so that applications can distinguish “failure to find a utility” from “invoked  
40532 utility exited with an error indication”. The value 127 was chosen because it is not commonly  
40533 used for other meanings; most utilities use small values for “normal error conditions” and the  
40534 values above 128 can be confused with termination due to receipt of a signal. The value 126 was  
40535 chosen in a similar manner to indicate that the utility could be found, but not invoked. Some  
40536 scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
40537 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
40538 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
40539 any other reason.

**40540 EXAMPLES**

- 40541 1. The following command combines the output of the parenthesised commands onto one  
40542 line, which is then written to the end-of-file **log**:

40543 `(logname; date; printf "%s\n" "$0 $*") | xargs >>log`

- 40544 2. The following command invokes *diff* with successive pairs of arguments originally typed  
40545 as command line arguments (assuming there are no embedded <blank>s in the elements of  
40546 the original argument list):

```
40547 printf "%s\n" "$*" | xargs -n 2 -x diff
40548 3. In the following commands, the user is asked which files in the current directory are to be
40549 archived. The files are archived into arch; a, one at a time, or b, many at a time.
40550 a. ls | xargs -p -L 1 ar -r arch
40551 b. ls | xargs -p -L 1 | xargs ar -r arch
40552 4. The following executes with successive pairs of arguments originally typed as command
40553 line arguments:
40554 echo $* | xargs -n 2 diff
40555 5. On XSI-conformant systems, the following moves all files from directory $1 to directory $2,
40556 and echoes each move command just before doing it:
40557 ls $1 | xargs -I {} -t mv ${1/{} ${2/{}}}
```

#### 40558 RATIONALE

40559 The *xargs* utility was usually found only in System V-based systems; BSD systems included an  
40560 *apply* utility that provided functionality similar to *xargs -n number*. The SVID lists *xargs* as a  
40561 software development extension. This volume of IEEE Std 1003.1-2001 does not share the view  
40562 that it is used only for development, and therefore it is not optional.

40563 The classic application of the *xargs* utility is in conjunction with the *find* utility to reduce the  
40564 number of processes launched by a simplistic use of the *find -exec* combination. The *xargs* utility  
40565 is also used to enforce an upper limit on memory required to launch a process. With this basis in  
40566 mind, this volume of IEEE Std 1003.1-2001 selected only the minimal features required.

40567 Although the 255 exit status is mostly an accident of historical implementations, it allows a  
40568 utility being used by *xargs* to tell *xargs* to terminate if it knows no further invocations using the  
40569 current data stream shall succeed. Any non-zero exit status from a utility falls into the 1-125  
40570 range when *xargs* exits. There is no statement of how the various non-zero utility exit status  
40571 codes are accumulated by *xargs*. The value could be the addition of all codes, their highest  
40572 value, the last one received, or a single value such as 1. Since no algorithm is arguably better  
40573 than the others, and since many of the standard utilities say little more (portably) than  
40574 "pass/fail", no new algorithm was invented.

40575 Several other *xargs* options were withdrawn because simple alternatives already exist within this  
40576 volume of IEEE Std 1003.1-2001. For example, the *-i replstr* option can be just as efficiently  
40577 performed using a shell **for** loop. Since *xargs* calls an *exec* function with each input line, the *-i*  
40578 option does not usually exploit the grouping capabilities of *xargs*.

40579 The requirement that *xargs* never produces command lines such that invocation of *utility* is  
40580 within 2 048 bytes of hitting the POSIX *exec {ARG\_MAX}* limitations is intended to guarantee  
40581 that the invoked utility has room to modify its environment variables and command line  
40582 arguments and still be able to invoke another utility. Note that the minimum *{ARG\_MAX}*  
40583 allowed by the System Interfaces volume of IEEE Std 1003.1-2001 is 4 096 bytes and the  
40584 minimum value allowed by this volume of IEEE Std 1003.1-2001 is 2 048 bytes; therefore, the  
40585 2 048 bytes difference seems reasonable. Note, however, that *xargs* may never be able to invoke a  
40586 utility if the environment passed in to *xargs* comes close to using *{ARG\_MAX}* bytes.

40587 The version of *xargs* required by this volume of IEEE Std 1003.1-2001 is required to wait for the  
40588 completion of the invoked command before invoking another command. This was done because  
40589 historical scripts using *xargs* assumed sequential execution. Implementations wanting to provide  
40590 parallel operation of the invoked utilities are encouraged to add an option enabling parallel  
40591 invocation, but should still wait for termination of all of the children before *xargs* terminates  
40592 normally.

40593     The **-e** option was omitted from the ISO POSIX-2:1993 standard in the belief that the *eofstr*  
40594     option-argument was recognized only when it was on a line by itself and before quote and  
40595     escape processing were performed, and that the logical end-of-file processing was only enabled  
40596     if a **-e** option was specified. In that case, a simple *sed* script could be used to duplicate the **-e**  
40597     functionality. Further investigation revealed that:

- 40598       • The logical end-of-file string was checked for after quote and escape processing, making a *sed*  
40599       script that provided equivalent functionality much more difficult to write.
- 40600       • The default was to perform logical end-of-file processing with an underscore as the logical  
40601       end-of-file string.

40602     To correct this misunderstanding, the **-E** *eofstr* option was adopted from the X/Open Portability  
40603     Guide. Users should note that the description of the **-E** option matches historical documentation  
40604     of the **-e** option (which was not adopted because it did not support the Utility Syntax  
40605     Guidelines), by saying that if *eofstr* is the null string, logical end-of-file processing is disabled.  
40606     Historical implementations of *xargs* actually did not disable logical end-of-file processing; they  
40607     treated a null argument found in the input as a logical end-of-file string. (A null *string* argument  
40608     could be generated using single or double quotes (' ' or " "). Since this behavior was not  
40609     documented historically, it is considered to be a bug.

## 40610 FUTURE DIRECTIONS

40611     None.

## 40612 SEE ALSO

40613     Chapter 2 (on page 29), *echo*, *find*, the System Interfaces volume of IEEE Std 1003.1-2001, *exec*

## 40614 CHANGE HISTORY

40615     First released in Issue 2.

## 40616 Issue 5

40617     A second FUTURE DIRECTION is added.

## 40618 Issue 6

40619     The obsolescent **-e**, **-i**, and **-l** options are removed.

40620     The following new requirements on POSIX implementations derive from alignment with the  
40621     Single UNIX Specification:

- 40622       • The **-p** option is added.
- 40623       • In the INPUT FILES section, the file **/dev/tty** is used to read responses required by the **-p**  
40624       option.
- 40625       • The STDERR section is updated to describe the **-p** option.

40626     The description of the **-E** option is aligned with the ISO POSIX-2:1993 standard.

40627     The normative text is reworded to avoid use of the term “must” for application requirements.

## 40628 NAME

40629        yacc — yet another compiler compiler (**DEVELOPMENT**)

## 40630 SYNOPSIS

40631 CD        `yacc [-dltv][-b file_prefix][-p sym_prefix] grammar`

40632

## 40633 DESCRIPTION

40634        The *yacc* utility shall read a description of a context-free grammar in *grammar* and write C source code, conforming to the ISO C standard, to a code file, and optionally header information into a header file, in the current directory. The C code shall define a function and related routines and macros for an automaton that executes a parsing algorithm meeting the requirements in **Algorithms** (on page 1071).

40639        The form and meaning of the grammar are described in the EXTENDED DESCRIPTION section.

40640        The C source code and header file shall be produced in a form suitable as input for the C compiler (see *c99*).

## 40642 OPTIONS

40643        The *yacc* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

40645        The following options shall be supported:

40646        **-b** *file\_prefix* Use *file\_prefix* instead of *y* as the prefix for all output filenames. The code file *y.tab.c*, the header file *y.tab.h* (created when **-d** is specified), and the description file *y.output* (created when **-v** is specified), shall be changed to *file\_prefix.tab.c*, *file\_prefix.tab.h*, and *file\_prefix.output*, respectively.

40650        **-d** Write the header file; by default only the code file is written. The **#define** statements associate the token codes assigned by *yacc* with the user-declared token names. This allows source files other than *y.tab.c* to access the token codes.

40653        **-l** Produce a code file that does not contain any **#line** constructs. If this option is not present, it is unspecified whether the code file or header file contains **#line** directives. This should only be used after the grammar and the associated actions are fully debugged.

40657        **-p** *sym\_prefix* Use *sym\_prefix* instead of *yy* as the prefix for all external names produced by *yacc*. The names affected shall include the functions *yparse()*, *yylex()*, and *yyerror()*, and the variables *yylval*, *ychar*, and *yydebug*. (In the remainder of this section, the six symbols cited are referenced using their default names only as a notational convenience.) Local names may also be affected by the **-p** option; however, the **-p** option shall not affect **#define** symbols generated by *yacc*.

40664        **-t** Modify conditional compilation directives to permit compilation of debugging code in the code file. Runtime debugging statements shall always be contained in the code file, but by default conditional compilation directives prevent their compilation.

40668        **-v** Write a file containing a description of the parser and a report of conflicts generated by ambiguities in the grammar.

**40670 OPERANDS**

40671 The following operand is required:

40672 *grammar* A pathname of a file containing instructions, hereafter called *grammar*, for which a parser is to be created. The format for the grammar is described in the EXTENDED DESCRIPTION section.

**40675 STDIN**

40676 Not used.

**40677 INPUT FILES**

40678 The file *grammar* shall be a text file formatted as specified in the EXTENDED DESCRIPTION section.

**40680 ENVIRONMENT VARIABLES**

40681 The following environment variables shall affect the execution of yacc:

40682 *LANG* Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

40686 *LC\_ALL* If set to a non-empty string value, override the values of all the other internationalization variables.

40688 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

**40691 *LC\_MESSAGES***

40692 Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

40694 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

40695 The *LANG* and *LC\_\** variables affect the execution of the yacc utility as stated. The *main()* function defined in **Yacc Library** (on page 1071) shall call:

40697 `setlocale(LC_ALL, "")`

40698 and thus the program generated by yacc shall also be affected by the contents of these variables at runtime.

**40700 ASYNCHRONOUS EVENTS**

40701 Default.

**40702 STDOUT**

40703 Not used.

**40704 STDERR**

40705 If shift/reduce or reduce/reduce conflicts are detected in *grammar*, yacc shall write a report of those conflicts to the standard error in an unspecified format.

40707 Standard error shall also be used for diagnostic messages.

**40708 OUTPUT FILES**

40709 The code file, the header file, and the description file shall be text files. All are described in the following sections.

40711

**Code File**40712  
40713  
40714  
40715  
40716

This file shall contain the C source code for the *yyparse()* function. It shall contain code for the various semantic actions with macro substitution performed on them as described in the EXTENDED DESCRIPTION section. It also shall contain a copy of the **#define** statements in the header file. If a **%union** declaration is used, the declaration for YYSTYPE shall also be included in this file.

40717

**Header File**40718  
40719  
40720  
40721

The header file shall contain **#define** statements that associate the token numbers with the token names. This allows source files other than the code file to access the token codes. If a **%union** declaration is used, the declaration for YYSTYPE and an *extern YYSTYPE yylval* declaration shall also be included in this file.

40722

**Description File**40723  
40724  
40725  
40726  
40727

The description file shall be a text file containing a description of the state machine corresponding to the parser, using an unspecified format. Limits for internal tables (see **Limits** (on page 1072)) shall also be reported, in an implementation-defined manner. (Some implementations may use dynamic allocation techniques and have no specific limit values to report.)

40728

**EXTENDED DESCRIPTION**40729  
40730  
40731

The *yacc* command accepts a language that is used to define a grammar for a target language to be parsed by the tables and code generated by *yacc*. The language accepted by *yacc* as a grammar for the target language is described below using the *yacc* input language itself.

40732  
40733  
40734  
40735  
40736  
40737

The input *grammar* includes rules describing the input structure of the target language and code to be invoked when these rules are recognized to provide the associated semantic action. The code to be executed shall appear as bodies of text that are intended to be C-language code. The C-language inclusions are presumed to form a correct function when processed by *yacc* into its output files. The code included in this way shall be executed during the recognition of the target language.

40738  
40739  
40740  
40741  
40742  
40743  
40744  
40745

Given a grammar, the *yacc* utility generates the files described in the OUTPUT FILES section. The code file can be compiled and linked using *c99*. If the declaration and programs sections of the grammar file did not include definitions of *main()*, *yylex()*, and *yyerror()*, the compiled output requires linking with externally supplied versions of those functions. Default versions of *main()* and *yyerror()* are supplied in the *yacc* library and can be linked in by using the *-ly* operand to *c99*. The *yacc* library interfaces need not support interfaces with other than the default *yy* symbol prefix. The application provides the lexical analyzer function, *yylex()*; the *lex* utility is specifically designed to generate such a routine.

40746

**Input Language**40747  
40748  
40749  
40750

The application shall ensure that every specification file consists of three sections in order: *declarations*, *grammar rules*, and *programs*, separated by double percent signs ("%%"). The declarations and programs sections can be empty. If the latter is empty, the preceding "%%" mark separating it from the rules section can be omitted.

40751

The input is free form text following the structure of the grammar defined below.

40752      **Lexical Structure of the Grammar**

40753      The <blank>s, <newline>s, and <form-feed>s shall be ignored, except that the application shall  
 40754      ensure that they do not appear in names or multi-character reserved symbols. Comments shall  
 40755      be enclosed in "/\* . . . \*/", and can appear wherever a name is valid.

40756      Names are of arbitrary length, made up of letters, periods ('.'), underscores ('\_'), and non-  
 40757      initial digits. Uppercase and lowercase letters are distinct. Conforming applications shall not  
 40758      use names beginning in yy or YY since the yacc parser uses such names. Many of the names  
 40759      appear in the final output of yacc, and thus they should be chosen to conform with any  
 40760      additional rules created by the C compiler to be used. In particular they appear in #define  
 40761      statements.

40762      A literal shall consist of a single character enclosed in single-quotes (''). All of the escape  
 40763      sequences supported for character constants by the ISO C standard shall be supported by yacc.

40764      The relationship with the lexical analyzer is discussed in detail below.

40765      The application shall ensure that the NUL character is not used in grammar rules or literals.

40766      **Declarations Section**

40767      The declarations section is used to define the symbols used to define the target language and  
 40768      their relationship with each other. In particular, much of the additional information required to  
 40769      resolve ambiguities in the context-free grammar for the target language is provided here.

40770      Usually yacc assigns the relationship between the symbolic names it generates and their  
 40771      underlying numeric value. The declarations section makes it possible to control the assignment  
 40772      of these values.

40773      It is also possible to keep semantic information associated with the tokens currently on the parse  
 40774      stack in a user-defined C-language **union**, if the members of the union are associated with the  
 40775      various names in the grammar. The declarations section provides for this as well.

40776      The first group of declarators below all take a list of names as arguments. That list can optionally  
 40777      be preceded by the name of a C union member (called a *tag* below) appearing within '<' and  
 40778      '>'. (As an exception to the typographical conventions of the rest of this volume of  
 40779      IEEE Std 1003.1-2001, in this case <*tag*> does not represent a metavariable, but the literal angle  
 40780      bracket characters surrounding a symbol.) The use of *tag* specifies that the tokens named on this  
 40781      line shall be of the same C type as the union member referenced by *tag*. This is discussed in  
 40782      more detail below.

40783      For lists used to define tokens, the first appearance of a given token can be followed by a  
 40784      positive integer (as a string of decimal digits). If this is done, the underlying value assigned to it  
 40785      for lexical purposes shall be taken to be that number.

40786      The following declares *name* to be a token:

40787      token [<*tag*>] *name* [*number*][*name* [*number*]]...

40788      If *tag* is present, the C type for all tokens on this line shall be declared to be the type referenced  
 40789      by *tag*. If a positive integer, *number*, follows a *name*, that value shall be assigned to the token.

40790      The following declares *name* to be a token, and assigns precedence to it:

40791      %left [<*tag*>] *name* [*number*][*name* [*number*]]...  
 40792      %right [<*tag*>] *name* [*number*][*name* [*number*]]...

40793      One or more lines, each beginning with one of these symbols, can appear in this section. All  
 40794      tokens on the same line have the same precedence level and associativity; the lines are in order

40795 of increasing precedence or binding strength. **%left** denotes that the operators on that line are  
40796 left associative, and **%right** similarly denotes right associative operators. If **tag** is present, it shall  
40797 declare a C type for *names* as described for **%token**.

40798 The following declares *name* to be a token, and indicates that this cannot be used associatively:

40799 **%nonassoc [<tag>] name [number][name [number]]...**

40800 If the parser encounters associative use of this token it reports an error. If **tag** is present, it shall  
40801 declare a C type for *names* as described for **%token**.

40802 The following declares that union member *names* are non-terminals, and thus it is required to  
40803 have a **tag** field at its beginning:

40804 **%type <tag> name...**

40805 Because it deals with non-terminals only, assigning a token number or using a literal is also  
40806 prohibited. If this construct is present, *yacc* shall perform type checking; if this construct is not  
40807 present, the parse stack shall hold only the **int** type.

40808 Every name used in *grammar* not defined by a **%token**, **%left**, **%right**, or **%nonassoc** declaration  
40809 is assumed to represent a non-terminal symbol. The *yacc* utility shall report an error for any  
40810 non-terminal symbol that does not appear on the left side of at least one grammar rule.

40811 Once the type, precedence, or token number of a name is specified, it shall not be changed. If the  
40812 first declaration of a token does not assign a token number, *yacc* shall assign a token number.  
40813 Once this assignment is made, the token number shall not be changed by explicit assignment.

40814 The following declarators do not follow the previous pattern.

40815 The following declares the non-terminal *name* to be the *start symbol*, which represents the largest,  
40816 most general structure described by the grammar rules:

40817 **%start name**

40818 By default, it is the left-hand side of the first grammar rule; this default can be overridden with  
40819 this declaration.

40820 The following declares the *yacc* value stack to be a union of the various types of values desired:

40821 **%union { body of union (in C) }**

40822 By default, the values returned by actions (see below) and the lexical analyzer shall be of type  
40823 **int**. The *yacc* utility keeps track of types, and it shall insert corresponding union member names  
40824 in order to perform strict type checking of the resulting parser.

40825 Alternatively, given that at least one **<tag>** construct is used, the union can be declared in a  
40826 header file (which shall be included in the declarations section by using a **#include** construct  
40827 within **%{** and **%}**), and a **typedef** used to define the symbol YYSTYPE to represent this union.  
40828 The effect of **%union** is to provide the declaration of YYSTYPE directly from the *yacc* input.

40829 C-language declarations and definitions can appear in the declarations section, enclosed by the  
40830 following marks:

40831 **%{ ... %}**

40832 These statements shall be copied into the code file, and have global scope within it so that they  
40833 can be used in the rules and program sections.

40834 The application shall ensure that the declarations section is terminated by the token **%%**.

40835      **Grammar Rules in yacc**

40836      The rules section defines the context-free grammar to be accepted by the function *yacc* generates,  
40837      and associates with those rules C-language actions and additional precedence information. The  
40838      grammar is described below, and a formal definition follows.

40839      The rules section is comprised of one or more grammar rules. A grammar rule has the form:

40840      A : BODY ;

40841      The symbol **A** represents a non-terminal name, and **BODY** represents a sequence of zero or  
40842      more *names*, *literals*, and *semantic actions* that can then be followed by optional *precedence rules*.  
40843      Only the names and literals participate in the formation of the grammar; the semantic actions  
40844      and precedence rules are used in other ways. The colon and the semicolon are *yacc* punctuation.  
40845      If there are several successive grammar rules with the same left-hand side, the vertical bar ' | '  
40846      can be used to avoid rewriting the left-hand side; in this case the semicolon appears only after  
40847      the last rule. The BODY part can be empty (or empty of names and literals) to indicate that the  
40848      non-terminal symbol matches the empty string.

40849      The *yacc* utility assigns a unique number to each rule. Rules using the vertical bar notation are  
40850      distinct rules. The number assigned to the rule appears in the description file.

40851      The elements comprising a BODY are:

40852      *name, literal*    These form the rules of the grammar: *name* is either a *token* or a *non-terminal*; *literal*  
40853      stands for itself (less the lexically required quotation marks).

40854      *semantic action*

40855      With each grammar rule, the user can associate actions to be performed each time  
40856      the rule is recognized in the input process. (Note that the word "action" can also  
40857      refer to the actions of the parser—shift, reduce, and so on.)

40858      These actions can return values and can obtain the values returned by previous  
40859      actions. These values are kept in objects of type YYSTYPE (see %union). The  
40860      result value of the action shall be kept on the parse stack with the left-hand side of  
40861      the rule, to be accessed by other reductions as part of their right-hand side. By  
40862      using the <tag> information provided in the declarations section, the code  
40863      generated by *yacc* can be strictly type checked and contain arbitrary information. In  
40864      addition, the lexical analyzer can provide the same kinds of values for tokens, if  
40865      desired.

40866      An action is an arbitrary C statement and as such can do input or output, call  
40867      subprograms, and alter external variables. An action is one or more C statements  
40868      enclosed in curly braces '{' and '}'.

40869      Certain pseudo-variables can be used in the action. These are macros for access to  
40870      data structures known internally to *yacc*.

40871      **\$\$**      The value of the action can be set by assigning it to **\$\$**. If type  
40872      checking is enabled and the type of the value to be assigned cannot  
40873      be determined, a diagnostic message may be generated.

40874      **\$number**    This refers to the value returned by the component specified by the  
40875      token *number* in the right side of a rule, reading from left to right;  
40876      *number* can be zero or negative. If *number* is zero or negative, it refers  
40877      to the data associated with the name on the parser's stack preceding  
40878      the leftmost symbol of the current rule. (That is, "\$0" refers to the  
40879      name immediately preceding the leftmost name in the current rule to  
40880      be found on the parser's stack and "\$-1" refers to the symbol to its

40881  
40882  
40883  
40884

left.) If *number* refers to an element past the current point in the rule, or beyond the bottom of the stack, the result is undefined. If type checking is enabled and the type of the value to be assigned cannot be determined, a diagnostic message may be generated.

40885       \$<tag>*number*  
40886  
40887  
40888  
40889  
40890

These correspond exactly to the corresponding symbols without the *tag* inclusion, but allow for strict type checking (and preclude unwanted type conversions). The effect is that the macro is expanded to use *tag* to select an element from the YYSTYPE union (using *dataname.tag*). This is particularly useful if *number* is not positive.

40891       \$<tag>\$  
40892  
40893  
40894

This imposes on the reference the type of the union member referenced by *tag*. This construction is applicable when a reference to a left context value occurs in the grammar, and provides yacc with a means for selecting a type.

40895  
40896  
40897  
40898  
40899  
40900  
40901  
40902

Actions can occur anywhere in a rule (not just at the end); an action can access values returned by actions to its left, and in turn the value it returns can be accessed by actions to its right. An action appearing in the middle of a rule shall be equivalent to replacing the action with a new non-terminal symbol and adding an empty rule with that non-terminal symbol on the left-hand side. The semantic action associated with the new rule shall be equivalent to the original action. The use of actions within rules might introduce conflicts that would not otherwise exist.

40903  
40904  
40905

By default, the value of a rule shall be the value of the first element in it. If the first element does not have a type (particularly in the case of a literal) and type checking is turned on by %type, an error message shall result.

40906       precedence  
40907  
40908  
40909  
40910  
40911  
40912  
40913

The keyword %prec can be used to change the precedence level associated with a particular grammar rule. Examples of this are in cases where a unary and binary operator have the same symbolic representation, but need to be given different precedences, or where the handling of an ambiguous if-else construction is necessary. The reserved symbol %prec can appear immediately after the body of the grammar rule and can be followed by a token name or a literal. It shall cause the precedence of the grammar rule to become that of the following token name or literal. The action for the rule as a whole can follow %prec.

40914  
40915

If a program section follows, the application shall ensure that the grammar rules are terminated by %%.

#### 40916       Programs Section

40917  
40918  
40919  
40920  
40921  
40922

The programs section can include the definition of the lexical analyzer yylex(), and any other functions; for example, those used in the actions specified in the grammar rules. It is unspecified whether the programs section precedes or follows the semantic actions in the output file; therefore, if the application contains any macro definitions and declarations intended to apply to the code in the semantic actions, it shall place them within "%{ . . . %}" in the declarations section.

**Input Grammar**

The following input to yacc yields a parser for the input to yacc. This formal syntax takes precedence over the preceding text syntax description.

The lexical structure is defined less precisely; **Lexical Structure of the Grammar** (on page 1063) defines most terms. The correspondence between the previous terms and the tokens below is as follows.

|                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40929<br>40930<br>40931<br>40932<br>40933<br>40934<br>40935<br>40936<br>40937<br>40938<br>40939<br>40940<br>40941<br>40942<br>40943<br>40944<br>40945<br>40946<br>40947<br>40948<br>40949<br>40950<br>40951<br>40952<br>40953<br>40954<br>40955<br>40956<br>40957<br>40958<br>40959<br>40960<br>40961<br>40962<br>40963<br>40964<br>40965<br>40966 | <b>IDENTIFIER</b> This corresponds to the concept of <i>name</i> , given previously. It also includes literals as defined previously.<br><b>C_IDENTIFIER</b> This is a name, and additionally it is known to be followed by a colon. A literal cannot yield this token.<br><b>NUMBER</b> A string of digits (a non-negative decimal integer).<br><b>TYPE, LEFT, MARK, LCURL, RCURL</b><br>These correspond directly to %type, %left, %% , %{, and %} .<br><b>{...}</b> This indicates C-language source code, with the possible inclusion of '\$' macros as discussed previously.<br><pre>/* Grammar for the input to yacc. */ /* Basic entries. */ /* The following are recognized by the lexical analyzer. */  %token IDENTIFIER      /* Includes identifiers and literals */ %token C_IDENTIFIER     /* identifier (but not literal)                            followed by a :. */ %token NUMBER          /* [0-9][0-9]* */  /* Reserved words : %type=&gt;TYPE %left=&gt;LEFT, and so on */  %token LEFT RIGHT NONASSOC TOKEN PREC TYPE START UNION  %token MARK             /* The %% mark. */ %token LCURL            /* The %{ mark. */ %token RCURL            /* The %} mark. */  /* 8-bit character literals stand for themselves; */ /* tokens have to be defined for multi-byte characters. */  %start spec  %%  spec   : defs MARK rules tail ; tail   : MARK {     /* In this action, set up the rest of the file. */ }   /* Empty; the second MARK is optional. */ ; defs   : /* Empty. */   defs def ; def    : START IDENTIFIER   UNION</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```
40967 {
40968 /* Copy union definition to output. */
40969 }
40970 | LCURL
40971 {
40972 /* Copy C code to output file. */
40973 }
40974 RCURL
40975 | rword tag nlist
40976 ;
40977 rword : TOKEN
40978 | LEFT
40979 | RIGHT
40980 | NONASSOC
40981 | TYPE
40982 ;
40983 tag : /* Empty: union tag ID optional. */
40984 | '<' IDENTIFIER '>'
40985 ;
40986 nlist : nmno
40987 | nlist nmno
40988 ;
40989 nmno : IDENTIFIER /* Note: literal invalid with % type. */
40990 | IDENTIFIER NUMBER /* Note: invalid with % type. */
40991 ;
40992 /* Rule section */
40993 rules : C_IDENTIFIER rbody prec
40994 | rules rule
40995 ;
40996 rule : C_IDENTIFIER rbody prec
40997 | '|' rbody prec
40998 ;
40999 rbody : /* empty */
41000 | rbody IDENTIFIER
41001 | rbody act
41002 ;
41003 act : '{'
41004 {
41005 /* Copy action, translate $$, and so on. */
41006 }
41007 '}'
41008 ;
41009 prec : /* Empty */
41010 | PREC IDENTIFIER
41011 | PREC IDENTIFIER act
41012 | prec ';'
41013 ;
```

41014

## Conflicts

41015  
41016  
41017  
41018

The parser produced for an input grammar may contain states in which conflicts occur. The conflicts occur because the grammar is not LALR(1). An ambiguous grammar always contains at least one LALR(1) conflict. The *yacc* utility shall resolve all conflicts, using either default rules or user-specified precedence rules.

41019  
41020  
41021  
41022

Conflicts are either shift/reduce conflicts or reduce/reduce conflicts. A shift/reduce conflict is where, for a given state and lookahead symbol, both a shift action and a reduce action are possible. A reduce/reduce conflict is where, for a given state and lookahead symbol, reductions by two different rules are possible.

41023  
41024  
41025  
41026  
41027  
41028

The rules below describe how to specify what actions to take when a conflict occurs. Not all shift/reduce conflicts can be successfully resolved this way because the conflict may be due to something other than ambiguity, so incautious use of these facilities can cause the language accepted by the parser to be much different from that which was intended. The description file shall contain sufficient information to understand the cause of the conflict. Where ambiguity is the reason either the default or explicit rules should be adequate to produce a working parser.

41029  
41030

The declared precedences and associativities (see **Declarations Section** (on page 1063)) are used to resolve parsing conflicts as follows:

41031  
41032  
41033  
41034

1. A precedence and associativity is associated with each grammar rule; it is the precedence and associativity of the last token or literal in the body of the rule. If the %prec keyword is used, it overrides this default. Some grammar rules might not have both precedence and associativity.

41035  
41036  
41037  
41038  
41039

2. If there is a shift/reduce conflict, and both the grammar rule and the input symbol have precedence and associativity associated with them, then the conflict is resolved in favor of the action (shift or reduce) associated with the higher precedence. If the precedences are the same, then the associativity is used; left associative implies reduce, right associative implies shift, and non-associative implies an error in the string being parsed.

41040  
41041  
41042

3. When there is a shift/reduce conflict that cannot be resolved by rule 2, the shift is done. Conflicts resolved this way are counted in the diagnostic output described in **Error Handling**.

41043  
41044  
41045

4. When there is a reduce/reduce conflict, a reduction is done by the grammar rule that occurs earlier in the input sequence. Conflicts resolved this way are counted in the diagnostic output described in **Error Handling**.

41046  
41047

Conflicts resolved by precedence or associativity shall not be counted in the shift/reduce and reduce/reduce conflicts reported by *yacc* on either standard error or in the description file.

41048

## Error Handling

41049  
41050  
41051  
41052

The token **error** shall be reserved for error handling. The name **error** can be used in grammar rules. It indicates places where the parser can recover from a syntax error. The default value of **error** shall be 256. Its value can be changed using a %token declaration. The lexical analyzer should not return the value of **error**.

41053  
41054  
41055  
41056

The parser shall detect a syntax error when it is in a state where the action associated with the lookahead symbol is **error**. A semantic action can cause the parser to initiate error handling by executing the macro YYERROR. When YYERROR is executed, the semantic action passes control back to the parser. YYERROR cannot be used outside of semantic actions.

41057  
41058

When the parser detects a syntax error, it normally calls *yyerror()* with the character string "syntax error" as its argument. The call shall not be made if the parser is still recovering

41059 from a previous error when the error is detected. The parser is considered to be recovering from  
41060 a previous error until the parser has shifted over at least three normal input symbols since the  
41061 last error was detected or a semantic action has executed the macro *yyerrok*. The parser shall not  
41062 call *yyerror()* when YYERROR is executed.

41063 The macro function YYRECOVERING shall return 1 if a syntax error has been detected and the  
41064 parser has not yet fully recovered from it. Otherwise, zero shall be returned.

41065 When a syntax error is detected by the parser, the parser shall check if a previous syntax error  
41066 has been detected. If a previous error was detected, and if no normal input symbols have been  
41067 shifted since the preceding error was detected, the parser checks if the lookahead symbol is an  
41068 endmarker (see **Interface to the Lexical Analyzer**). If it is, the parser shall return with a non-  
41069 zero value. Otherwise, the lookahead symbol shall be discarded and normal parsing shall  
41070 resume.

41071 When YYERROR is executed or when the parser detects a syntax error and no previous error has  
41072 been detected, or at least one normal input symbol has been shifted since the previous error was  
41073 detected, the parser shall pop back one state at a time until the parse stack is empty or the  
41074 current state allows a shift over **error**. If the parser empties the parse stack, it shall return with a  
41075 non-zero value. Otherwise, it shall shift over **error** and then resume normal parsing. If the parser  
41076 reads a lookahead symbol before the error was detected, that symbol shall still be the lookahead  
41077 symbol when parsing is resumed.

41078 The macro *yyerrok* in a semantic action shall cause the parser to act as if it has fully recovered  
41079 from any previous errors. The macro *yyclearin* shall cause the parser to discard the current  
41080 lookahead token. If the current lookahead token has not yet been read, *yyclearin* shall have no  
41081 effect.

41082 The macro YYACCEPT shall cause the parser to return with the value zero. The macro  
41083 YYABORT shall cause the parser to return with a non-zero value.

#### 41084 **Interface to the Lexical Analyzer**

41085 The *yylex()* function is an integer-valued function that returns a *token number* representing the  
41086 kind of token read. If there is a value associated with the token returned by *yylex()* (see the  
41087 discussion of *tag* above), it shall be assigned to the external variable *yylval*.

41088 If the parser and *yylex()* do not agree on these token numbers, reliable communication between  
41089 them cannot occur. For (single-byte character) literals, the token is simply the numeric value of  
41090 the character in the current character set. The numbers for other tokens can either be chosen by  
41091 *yacc*, or chosen by the user. In either case, the **#define** construct of C is used to allow *yylex()* to  
41092 return these numbers symbolically. The **#define** statements are put into the code file, and the  
41093 header file if that file is requested. The set of characters permitted by *yacc* in an identifier is larger  
41094 than that permitted by C. Token names found to contain such characters shall not be included in  
41095 the **#define** declarations.

41096 If the token numbers are chosen by *yacc*, the tokens other than literals shall be assigned numbers  
41097 greater than 256, although no order is implied. A token can be explicitly assigned a number by  
41098 following its first appearance in the declarations section with a number. Names and literals not  
41099 defined this way retain their default definition. All token numbers assigned by *yacc* shall be  
41100 unique and distinct from the token numbers used for literals and user-assigned tokens. If  
41101 duplicate token numbers cause conflicts in parser generation, *yacc* shall report an error;  
41102 otherwise, it is unspecified whether the token assignment is accepted or an error is reported.

41103 The end of the input is marked by a special token called the *endmarker*, which has a token  
41104 number that is zero or negative. (These values are invalid for any other token.) All lexical  
41105 analyzers shall return zero or negative as a token number upon reaching the end of their input. If

41106 the tokens up to, but excluding, the endmarker form a structure that matches the start symbol,  
41107 the parser shall accept the input. If the endmarker is seen in any other context, it shall be  
41108 considered an error.

41109 **Completing the Program**

41110 In addition to *yyparse()* and *yylex()*, the functions *yyerror()* and *main()* are required to make a  
41111 complete program. The application can supply *main()* and *yyerror()*, or those routines can be  
41112 obtained from the *yacc* library.

41113 **Yacc Library**

41114 The following functions shall appear only in the *yacc* library accessible through the **-ly** operand  
41115 to *c99*; they can therefore be redefined by a conforming application:

41116 **int main(void)**

41117 This function shall call *yyparse()* and exit with an unspecified value. Other actions within  
41118 this function are unspecified.

41119 **int yyerror(const char \*s)**

41120 This function shall write the NUL-terminated argument to standard error, followed by a  
41121 <newline>.

41122 The order of the **-ly** and **-lI** operands given to *c99* is significant; the application shall either  
41123 provide its own *main()* function or ensure that **-ly** precedes **-lI**.

41124 **Debugging the Parser**

41125 The parser generated by *yacc* shall have diagnostic facilities in it that can be optionally enabled  
41126 at either compile time or at runtime (if enabled at compile time). The compilation of the runtime  
41127 debugging code is under the control of YYDEBUG, a preprocessor symbol. If YYDEBUG has a  
41128 non-zero value, the debugging code shall be included. If its value is zero, the code shall not be  
41129 included.

41130 In parsers where the debugging code has been included, the external **int yydebug** can be used to  
41131 turn debugging on (with a non-zero value) and off (zero value) at runtime. The initial value of  
41132 *yydebug* shall be zero.

41133 When **-t** is specified, the code file shall be built such that, if YYDEBUG is not already defined at  
41134 compilation time (using the *c99 -D YYDEBUG* option, for example), YYDEBUG shall be set  
41135 explicitly to 1. When **-t** is not specified, the code file shall be built such that, if YYDEBUG is not  
41136 already defined, it shall be set explicitly to zero.

41137 The format of the debugging output is unspecified but includes at least enough information to  
41138 determine the shift and reduce actions, and the input symbols. It also provides information  
41139 about error recovery.

41140 **Algorithms**

41141 The parser constructed by *yacc* implements an LALR(1) parsing algorithm as documented in the  
41142 literature. It is unspecified whether the parser is table-driven or direct-coded.

41143 A parser generated by *yacc* shall never request an input symbol from *yylex()* while in a state  
41144 where the only actions other than the error action are reductions by a single rule.

41145 The literature of parsing theory defines these concepts.

41146

**Limits**

41147

The *yacc* utility may have several internal tables. The minimum maximums for these tables are shown in the following table. The exact meaning of these values is implementation-defined. The implementation shall define the relationship between these values and between them and any error messages that the implementation may generate should it run out of space for any internal structure. An implementation may combine groups of these resources into a single pool as long as the total available to the user does not fall below the sum of the sizes specified by this section.

41153

**Table 4-22 Internal Limits in yacc**

41154

| <b>Limit</b> | <b>Minimum<br/>Maximum</b> | <b>Description</b>                                                                                                                                                                                                                                        |
|--------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {INTERMS}    | 126                        | Number of tokens.                                                                                                                                                                                                                                         |
| {NNONTERM}   | 200                        | Number of non-terminals.                                                                                                                                                                                                                                  |
| {NPROD}      | 300                        | Number of rules.                                                                                                                                                                                                                                          |
| {NSTATES}    | 600                        | Number of states.                                                                                                                                                                                                                                         |
| {MEMSIZE}    | 5 200                      | Length of rules. The total length, in names (tokens and non-terminals), of all the rules of the grammar. The left-hand side is counted for each rule, even if it is not explicitly repeated, as specified in <b>Grammar Rules in yacc</b> (on page 1065). |
| {ACTSIZE}    | 4 000                      | Number of actions. “Actions” here (and in the description file) refer to parser actions (shift, reduce, and so on) not to semantic actions defined in <b>Grammar Rules in yacc</b> (on page 1065).                                                        |

**41171 EXIT STATUS**

41172

The following exit values shall be returned:

41173

0 Successful completion.

41174

>0 An error occurred.

**41175 CONSEQUENCES OF ERRORS**

41176

If any errors are encountered, the run is aborted and *yacc* exits with a non-zero status. Partial code files and header files may be produced. The summary information in the description file shall always be produced if the **-v** flag is present.

**41179 APPLICATION USAGE**

41180

Historical implementations experience name conflicts on the names **yacc.tmp**, **yacc.acts**, **yacc.debug**, **y.tab.c**, **y.tab.h**, and **y.output** if more than one copy of *yacc* is running in a single directory at one time. The **-b** option was added to overcome this problem. The related problem of allowing multiple *yacc* parsers to be placed in the same file was addressed by adding a **-p** option to override the previously hard-coded yy variable prefix.

41185

The description of the **-p** option specifies the minimal set of function and variable names that cause conflict when multiple parsers are linked together. YYSTYPE does not need to be changed. Instead, the programmer can use **-b** to give the header files for different parsers different names, and then the file with the *yylex()* for a given parser can include the header for that parser. Names such as *yyclearerr* do not need to be changed because they are used only in the actions; they do not have linkage. It is possible that an implementation has other names, either internal ones for implementing things such as *yyclearerr*, or providing non-standard features that it wants to change with **-p**.

41193      Unary operators that are the same token as a binary operator in general need their precedence  
41194      adjusted. This is handled by the **%prec** advisory symbol associated with the particular grammar  
41195      rule defining that unary operator. (See **Grammar Rules in yacc** (on page 1065).) Applications  
41196      are not required to use this operator for unary operators, but the grammars that do not require it  
41197      are rare.

#### 41198 EXAMPLES

41199      Access to the *yacc* library is obtained with library search operands to *c99*. To use the *yacc* library  
41200      *main()*:

41201      *c99 y.tab.c -l y*

41202      Both the *lex* library and the *yacc* library contain *main()*. To access the *yacc main()*:

41203      *c99 y.tab.c lex.yy.c -l y -l 1*

41204      This ensures that the *yacc* library is searched first, so that its *main()* is used.

41205      The historical *yacc* libraries have contained two simple functions that are normally coded by the  
41206      application programmer. These functions are similar to the following code:

```
41207 #include <locale.h>
41208 int main(void)
41209 {
41210 extern int yyparse();
41211
41212 setlocale(LC_ALL, "");
41213
41214 /* If the following parser is one created by lex, the
41215 application must be careful to ensure that LC_CTYPE
41216 and LC_COLLATE are set to the POSIX locale. */
41217 (void) yyparse();
41218 return (0);
41219 }
41220
41221 #include <stdio.h>
41222
41223 int yyerror(const char *msg)
41224 {
41225 (void) fprintf(stderr, "%s\n", msg);
41226 return (0);
41227 }
```

#### 41224 RATIONALE

41225      The references in **Referenced Documents** (on page xxviii) may be helpful in constructing the  
41226      parser generator. The referenced DeRemer and Pennello article (along with the works it  
41227      references) describes a technique to generate parsers that conform to this volume of  
41228      IEEE Std 1003.1-2001. Work in this area continues to be done, so implementors should consult  
41229      current literature before doing any new implementations. The original Knuth article is the  
41230      theoretical basis for this kind of parser, but the tables it generates are impractically large for  
41231      reasonable grammars and should not be used. The “equivalent to” wording is intentional to  
41232      assure that the best tables that are LALR(1) can be generated.

41233      There has been confusion between the class of grammars, the algorithms needed to generate  
41234      parsers, and the algorithms needed to parse the languages. They are all reasonably orthogonal.  
41235      In particular, a parser generator that accepts the full range of LR(1) grammars need not generate  
41236      a table any more complex than one that accepts SLR(1) (a relatively weak class of LR grammars)  
41237      for a grammar that happens to be SLR(1). Such an implementation need not recognize the case,  
41238      either; table compression can yield the SLR(1) table (or one even smaller than that) without

41239 recognizing that the grammar is SLR(1). The speed of an LR(1) parser for any class is dependent  
41240 more upon the table representation and compression (or the code generation if a direct parser is  
41241 generated) than upon the class of grammar that the table generator handles.

41242 The speed of the parser generator is somewhat dependent upon the class of grammar it handles.  
41243 However, the original Knuth article algorithms for constructing LR parsers were judged by its  
41244 author to be impractically slow at that time. Although full LR is more complex than LALR(1), as  
41245 computer speeds and algorithms improve, the difference (in terms of acceptable wall-clock  
41246 execution time) is becoming less significant.

41247 Potential authors are cautioned that the referenced DeRemer and Pennello article previously  
41248 cited identifies a bug (an over-simplification of the computation of LALR(1) lookahead sets) in  
41249 some of the LALR(1) algorithm statements that preceded it to publication. They should take the  
41250 time to seek out that paper, as well as current relevant work, particularly Aho's.

41251 The **-b** option was added to provide a portable method for permitting yacc to work on multiple  
41252 separate parsers in the same directory. If a directory contains more than one yacc grammar, and  
41253 both grammars are constructed at the same time (by, for example, a parallel make program),  
41254 conflict results. While the solution is not historical practice, it corrects a known deficiency in  
41255 historical implementations. Corresponding changes were made to all sections that referenced  
41256 the filenames **y.tab.c** (now "the code file"), **y.tab.h** (now "the header file"), and **y.output** (now  
41257 "the description file").

41258 The grammar for yacc input is based on System V documentation. The textual description shows  
41259 there that the ';' is required at the end of the rule. The grammar and the implementation do not  
41260 require this. (The use of **C\_IDENTIFIER** causes a reduce to occur in the right place.)

41261 Also, in that implementation, the constructs such as **%token** can be terminated by a semicolon,  
41262 but this is not permitted by the grammar. The keywords such as **%token** can also appear in  
41263 uppercase, which is again not discussed. In most places where '%' is used, '\' can be  
41264 substituted, and there are alternate spellings for some of the symbols (for example, **%LEFT** can  
41265 be "%<" or even "\<").

41266 Historically, <tag> can contain any characters except '>', including white space, in the  
41267 implementation. However, since the tag must reference an ISO C standard union member, in  
41268 practice conforming implementations need to support only the set of characters for ISO C  
41269 standard identifiers in this context.

41270 Some historical implementations are known to accept actions that are terminated by a period.  
41271 Historical implementations often allow '\$' in names. A conforming implementation does not  
41272 need to support either of these behaviors.

41273 Deciding when to use **%prec** illustrates the difficulty in specifying the behavior of yacc. There  
41274 may be situations in which the grammar is not, strictly speaking, in error, and yet yacc cannot  
41275 interpret it unambiguously. The resolution of ambiguities in the grammar can in many instances  
41276 be resolved by providing additional information, such as using **%type** or **%union** declarations. It  
41277 is often easier and it usually yields a smaller parser to take this alternative when it is  
41278 appropriate.

41279 The size and execution time of a program produced without the runtime debugging code is  
41280 usually smaller and slightly faster in historical implementations.

41281 Statistics messages from several historical implementations include the following types of  
41282 information:

41283 n/512 terminals, n/300 non-terminals  
41284 n/600 grammar rules, n/1500 states  
41285 n shift/reduce, n reduce/reduce conflicts reported

41286        *n*/350 working sets used  
41287        Memory: states, etc. *n*/15 000, parser *n*/15 000  
41288        *n*/600 distinct lookahead sets  
41289        *n* extra closures  
41290        *n* shift entries, *n* exceptions  
41291        *n* goto entries  
41292        *n* entries saved by goto default  
41293        Optimizer space used: input *n*/15 000, output *n*/15 000  
41294        *n* table entries, *n* zero  
41295        Maximum spread: *n*, Maximum offset: *n*

41296        The report of internal tables in the description file is left implementation-defined because all  
41297        aspects of these limits are also implementation-defined. Some implementations may use  
41298        dynamic allocation techniques and have no specific limit values to report.

41299        The format of the **y.output** file is not given because specification of the format was not seen to  
41300        enhance applications portability. The listing is primarily intended to help human users  
41301        understand and debug the parser; use of **y.output** by a conforming application script would be  
41302        unusual. Furthermore, implementations have not produced consistent output and no popular  
41303        format was apparent. The format selected by the implementation should be human-readable, in  
41304        addition to the requirement that it be a text file.

41305        Standard error reports are not specifically described because they are seldom of use to  
41306        conforming applications and there was no reason to restrict implementations.

41307        Some implementations recognize "={ " as equivalent to '{ ' because it appears in historical  
41308        documentation. This construction was recognized and documented as obsolete as long ago as  
41309        1978, in the referenced *Yacc: Yet Another Compiler-Compiler*. This volume of IEEE Std 1003.1-2001  
41310        chose to leave it as obsolete and omit it.

41311        Multi-byte characters should be recognized by the lexical analyzer and returned as tokens. They  
41312        should not be returned as multi-byte character literals. The token **error** that is used for error  
41313        recovery is normally assigned the value 256 in the historical implementation. Thus, the token  
41314        value 256, which is used in many multi-byte character sets, is not available for use as the value  
41315        of a user-defined token.

#### 41316 FUTURE DIRECTIONS

41317        None.

#### 41318 SEE ALSO

41319        *c99*, *lex*

#### 41320 CHANGE HISTORY

41321        First released in Issue 2.

#### 41322 Issue 5

41323        The FUTURE DIRECTIONS section is added.

#### 41324 Issue 6

41325        This utility is marked as part of the C-Language Development Utilities option.

41326        Minor changes have been added to align with the IEEE P1003.2b draft standard.

41327        The normative text is reworded to avoid use of the term “must” for application requirements.

41328        IEEE PASC Interpretation 1003.2 #177 is applied, changing the comment on **RCURL** from the }%  
41329        token to the %}.

**41330 NAME**

41331 zcat — expand and concatenate data

**41332 SYNOPSIS**

41333 XSI zcat [*file...*]

41334

**41335 DESCRIPTION**

41336 The *zcat* utility shall write to standard output the uncompressed form of files that have been  
41337 compressed using the *compress* utility. It is the equivalent of *uncompress -c*. Input files are not  
41338 affected.

**41339 OPTIONS**

41340 None.

**41341 OPERANDS**

41342 The following operand shall be supported:

41343 *file* The pathname of a file previously processed by the *compress* utility. If *file* already  
41344 has the **.Z** suffix specified, it is used as submitted. Otherwise, the **.Z** suffix is  
41345 appended to the filename prior to processing.

**41346 STDIN**

41347 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '**-**'.

**41348 INPUT FILES**

41349 Input files shall be compressed files that are in the format produced by the *compress* utility.

**41350 ENVIRONMENT VARIABLES**

41351 The following environment variables shall affect the execution of *zcat*:

41352 *LANG* Provide a default value for the internationalization variables that are unset or null.  
41353 (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2,  
41354 Internationalization Variables for the precedence of internationalization variables  
41355 used to determine the values of locale categories.)

41356 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
41357 internationalization variables.

41358 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
41359 characters (for example, single-byte as opposed to multi-byte characters in  
41360 arguments).

**41361 *LC\_MESSAGES***

41362 Determine the locale that should be used to affect the format and contents of  
41363 diagnostic messages written to standard error.

41364 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**41365 ASYNCHRONOUS EVENTS**

41366 Default.

**41367 STDOUT**

41368 The compressed files given as input shall be written on standard output in their uncompressed  
41369 form.

**41370 STDERR**

41371 The standard error shall be used only for diagnostic messages.

**41372 OUTPUT FILES**

41373 None.

**41374 EXTENDED DESCRIPTION**

41375 None.

**41376 EXIT STATUS**

41377 The following exit values shall be returned:

41378 0 Successful completion.

41379 >0 An error occurred.

**41380 CONSEQUENCES OF ERRORS**

41381 Default.

**41382 APPLICATION USAGE**

41383 None.

**41384 EXAMPLES**

41385 None.

**41386 RATIONALE**

41387 None.

**41388 FUTURE DIRECTIONS**

41389 None.

**41390 SEE ALSO**

41391 *compress, uncompress*

**41392 CHANGE HISTORY**

41393 First released in Issue 4.



# Index

|                                           |             |                                       |          |
|-------------------------------------------|-------------|---------------------------------------|----------|
| <control>-V.....                          | 366         | string functions .....                | 167      |
| <control>-W .....                         | 366         | user-defined functions .....          | 169      |
| _CFLAGS.....                              | 215         | variables and special variables.....  | 160      |
| _LDFLAGS.....                             | 215         | background work                       |          |
| _LIBS .....                               | 215         | at .....                              | 144      |
| _POSIX_VDISABLE .....                     | 889         | batch.....                            | 190      |
| Account_Name .....                        | 108         | bg .....                              | 208      |
| actions equivalent to functions.....      | 7           | crontab.....                          | 273      |
| admin .....                               | 126         | fg .....                              | 440      |
| ADV .....                                 | 9           | jobs.....                             | 518      |
| AIO .....                                 | 9           | nice .....                            | 661      |
| alias .....                               | 28, 48, 131 | nohup.....                            | 673      |
| alias substitution.....                   | 32          | renice.....                           | 816      |
| AND lists.....                            | 51          | BAR.....                              | 9        |
| AND-OR list .....                         | 50          | basename.....                         | 187      |
| appending redirected output .....         | 44          | batch.....                            | 190      |
| ar .....                                  | 134         | batch administration.....             | 104      |
| archives                                  |             | batch authorization .....             | 103      |
| ar command.....                           | 134         | batch client-server interaction ..... | 101      |
| ARG_MAX.....                              | 1058        | batch environment                     |          |
| arithmetic expansion .....                | 41          | services .....                        | 101      |
| arithmetic language                       |             | utilities .....                       | 101      |
| bc.....                                   | 193         | Batch Job Abort.....                  | 103, 114 |
| arithmetic precision and operations ..... | 7           | batch job creation.....               | 102      |
| array identifiers.....                    | 198         | Batch Job Execution .....             | 103, 106 |
| asa .....                                 | 141         | Batch Job Exit .....                  | 103, 113 |
| asynchronous lists .....                  | 50          | batch job identifier.....             | 122      |
| at .....                                  | 144         | Batch Job Message Request .....       | 116      |
| at-job.....                               | 144         | Batch Job Routing.....                | 102, 113 |
| automatic storage class .....             | 202         | batch job states .....                | 105      |
| awk .....                                 | 153         | Batch Job Status Request .....        | 117      |
| actions.....                              | 164         | batch job tracking .....              | 102      |
| arithmetic functions.....                 | 166         | batch notification .....              | 104      |
| escape sequences.....                     | 162         | batch queue .....                     | 101      |
| expression patterns .....                 | 164         | Batch Queue Status Request .....      | 119      |
| expressions .....                         | 156         | Batch Server Restart .....            | 114      |
| functions.....                            | 166         | batch services .....                  | 104      |
| grammar.....                              | 170         | bc.....                               | 193      |
| input/output and general functions .....  | 168         | grammar .....                         | 194      |
| lexical conventions.....                  | 176         | lexical conventions.....              | 196      |
| output statements .....                   | 165         | operations .....                      | 198      |
| overall program structure .....           | 155         | operators .....                       | 198      |
| pattern ranges .....                      | 164         | bcc (mailer blind carbon copy) .....  | 605      |
| patterns.....                             | 163         | BC_BASE_MAX .....                     | 18       |
| regular expressions .....                 | 161         | BC_DIM_MAX .....                      | 18       |
| special patterns .....                    | 163         | BC_SCALE_MAX .....                    | 18       |

|                                             |             |  |
|---------------------------------------------|-------------|--|
| BC_STRING_MAX.....                          | 18, 196     |  |
| BE.....                                     | <b>10</b>   |  |
| bg.....                                     | 28, 48, 208 |  |
| binary primaries.....                       | 905         |  |
| break special built-in.....                 | <b>65</b>   |  |
| built-in utilities.....                     | 28          |  |
| builtin.....                                | 261         |  |
| c99.....                                    | <b>211</b>  |  |
| external symbols.....                       | 215         |  |
| standard libraries.....                     | 214         |  |
| cal.....                                    | <b>220</b>  |  |
| can.....                                    | 1           |  |
| carriage-control characters.....            | 141         |  |
| case conditional construct.....             | 53          |  |
| cat.....                                    | <b>222</b>  |  |
| cc (mailer carbon copy).....                | 605         |  |
| CD.....                                     | <b>10</b>   |  |
| cd .....                                    | 28, 48, 226 |  |
| cflow.....                                  | <b>230</b>  |  |
| changing the current working directory..... | 7           |  |
| character counting.....                     | 1042        |  |
| charmap                                     |             |  |
| with localedef.....                         | 555         |  |
| writing names with locale .....             | 550         |  |
| charmap file.....                           | 554, 889    |  |
| CHAR_BIT.....                               | 346         |  |
| Checkpoint.....                             | 108         |  |
| checksums                                   |             |  |
| cksum.....                                  | 246         |  |
| chgrp .....                                 | <b>233</b>  |  |
| chmod .....                                 | <b>236</b>  |  |
| grammar.....                                | 239         |  |
| chown.....                                  | <b>242</b>  |  |
| cksum.....                                  | <b>246</b>  |  |
| cmp .....                                   | <b>251</b>  |  |
| codeset conversion.....                     | 503         |  |
| tr .....                                    | 923         |  |
| COLL_WEIGHTS_MAX.....                       | 18          |  |
| colon special built-in.....                 | 67, 69      |  |
| comm.....                                   | <b>254</b>  |  |
| command .....                               | 28, 48, 257 |  |
| command mode.....                           | 334         |  |
| command search and execution.....           | 48          |  |
| command substitution .....                  | 40          |  |
| communications commands                     |             |  |
| mailx.....                                  | 583         |  |
| talk.....                                   | 897         |  |
| uucp.....                                   | 962         |  |
| uudecode.....                               | 966         |  |
| uuencode.....                               | 969         |  |
| uustat .....                                | 974         |  |
| uux.....                                    | 977         |  |
| write .....                                 | 1051        |  |
| compilers                                   |             |  |
| c99.....                                    | 211         |  |
| fort77 .....                                | 461         |  |
| yacc.....                                   | 1060        |  |
| compound commands.....                      | 52          |  |
| compound-list.....                          | 50          |  |
| compress.....                               | <b>262</b>  |  |
| compression                                 |             |  |
| compress .....                              | 262         |  |
| uncompress .....                            | 948         |  |
| zcat .....                                  | 1076        |  |
| concurrent execution of processes.....      | 3           |  |
| configuration values .....                  | 481         |  |
| conforming application.....                 | 139         |  |
| consequences of shell errors.....           | 46          |  |
| continue .....                              | <b>69</b>   |  |
| control characters .....                    | 886         |  |
| controlling terminal.....                   | 3           |  |
| Coordinated Universal Time (UTC).....       | 298         |  |
| copy files commands                         |             |  |
| cp.....                                     | 265         |  |
| dd.....                                     | 300         |  |
| ln .....                                    | 546         |  |
| mv .....                                    | 652         |  |
| pax .....                                   | 698         |  |
| cp.....                                     | <b>265</b>  |  |
| cpio format.....                            | 718         |  |
| CPT.....                                    | <b>10</b>   |  |
| CPU time .....                              | 914         |  |
| cron daemon.....                            | 276         |  |
| crontab .....                               | <b>273</b>  |  |
| CS.....                                     | <b>10</b>   |  |
| csplit .....                                | <b>277</b>  |  |
| ctags.....                                  | <b>281</b>  |  |
| current working directory.....              | 3           |  |
| cut .....                                   | <b>286</b>  |  |
| CX.....                                     | <b>10</b>   |  |
| cxref .....                                 | <b>290</b>  |  |
| data keywords.....                          | 744         |  |
| date .....                                  | <b>293</b>  |  |
| conversion specifications.....              | 293         |  |
| modified conversion specifications .....    | 294         |  |
| dd .....                                    | <b>300</b>  |  |
| default queue.....                          | 118         |  |
| deferred batch services.....                | 106         |  |
| Delete Batch Job Request .....              | 115         |  |
| delta .....                                 | <b>309</b>  |  |
| destination .....                           | 123         |  |
| df .....                                    | <b>313</b>  |  |

|                                              |     |                                    |          |
|----------------------------------------------|-----|------------------------------------|----------|
| diff.....                                    | 317 | read command .....                 | 342      |
| binary output format.....                    | 319 | regular expressions .....          | 335      |
| default output format.....                   | 319 | shell escape command .....         | 344      |
| directory comparison format.....             | 318 | substitute command.....            | 342      |
| -c or -C output format .....                 | 320 | undo command.....                  | 343      |
| -e output format.....                        | 320 | write command.....                 | 344      |
| -f output format .....                       | 320 | edit buffer.....                   | 353, 984 |
| directory commands                           |     | edit line .....                    | 852      |
| cd.....                                      | 226 | editors                            |          |
| pwd .....                                    | 755 | ed .....                           | 334      |
| directory lister .....                       | 568 | ex.....                            | 353      |
| dirname.....                                 | 324 | sed.....                           | 838      |
| disk space commands                          |     | vi .....                           | 984      |
| df.....                                      | 313 | ED_FILE_MAX.....                   | 346      |
| du .....                                     | 327 | ED_LINE_MAX .....                  | 346      |
| ulimit.....                                  | 937 | effective group ID .....           | 3        |
| documentation.....                           | 628 | effective user ID .....            | 3        |
| dot special built-in.....                    | 71  | Eighth Edition UNIX.....           | 261      |
| double-quotes.....                           | 30  | env .....                          | 350      |
| du .....                                     | 327 | EPERM.....                         | 270      |
| duplicating an input file descriptor .....   | 45  | Error_Path .....                   | 109      |
| duplicating an output file descriptor .....  | 45  | escape character (backslash) ..... | 30       |
| echo.....                                    | 331 | escape sequences                   |          |
| ed.....                                      | 334 | awk .....                          | 162      |
| addresses.....                               | 336 | gencat.....                        | 471      |
| append command .....                         | 338 | lex .....                          | 537      |
| change command .....                         | 338 | establish the locale.....          | 7        |
| commands.....                                | 337 | eval special built-in .....        | 73       |
| copy command .....                           | 343 | ex.....                            | 353      |
| delete command .....                         | 339 | <backslash> .....                  | 365      |
| edit command .....                           | 339 | <control>-D command.....           | 388      |
| edit without checking command .....          | 339 | <newline> .....                    | 365      |
| filename command .....                       | 339 | abbreviate command .....           | 368      |
| global command .....                         | 339 | addressing.....                    | 359      |
| global non-matched command .....             | 344 | adjust window command .....        | 386      |
| help command .....                           | 340 | append command .....               | 369      |
| help-mode command .....                      | 340 | args command .....                 | 369      |
| insert command .....                         | 340 | autoindent option .....            | 390      |
| interactive global command .....             | 340 | autoprint option .....             | 391      |
| interactive global not-matched command ..... | 344 | autowrite option .....             | 391      |
| join command .....                           | 341 | beautify option .....              | 391      |
| line number command .....                    | 344 | change command .....               | 369      |
| list command .....                           | 341 | chdir command .....                | 370      |
| mark command .....                           | 341 | command descriptions .....         | 366      |
| move command .....                           | 341 | copy command .....                 | 370      |
| null command .....                           | 345 | delete command .....               | 370      |
| number command .....                         | 341 | directory option .....             | 391      |
| print command .....                          | 342 | edcompatible option .....          | 391      |
| prompt command .....                         | 342 | edit command .....                 | 370      |
| quit command .....                           | 342 | edit options .....                 | 390      |
| quit without checking command .....          | 342 | errorbells option .....            | 392      |

|                                         |                    |
|-----------------------------------------|--------------------|
| escape command.....                     | 387                |
| execute command .....                   | 389                |
| exrc command.....                       | 392                |
| file command.....                       | 371                |
| global command.....                     | 372                |
| ignorecase option .....                 | 392                |
| initialization.....                     | 356                |
| input editing.....                      | 364                |
| insert command.....                     | 373                |
| join command .....                      | 373                |
| list command .....                      | 374, 392           |
| magic command.....                      | 392                |
| map command.....                        | 374                |
| mark command.....                       | 375                |
| mesg command.....                       | 392                |
| move command.....                       | 376                |
| next command .....                      | 376                |
| number command.....                     | 377                |
| number option .....                     | 393                |
| open command .....                      | 377                |
| paragraphs option.....                  | 393                |
| preserve .....                          | 353                |
| preserve command .....                  | 377                |
| print command .....                     | 378                |
| prompt command .....                    | 393                |
| put command.....                        | 378                |
| quit command .....                      | 378                |
| read command .....                      | 379                |
| readonly command .....                  | 393                |
| recover command.....                    | 379                |
| redraw command.....                     | 394                |
| regular expressions .....               | 389                |
| remap command .....                     | 394                |
| replacement strings.....                | 389                |
| report command.....                     | 394                |
| rewind command .....                    | 380                |
| scroll command .....                    | 364, 394           |
| sections command.....                   | 394                |
| set command .....                       | 380                |
| shell command.....                      | 380                |
| shell option .....                      | 395                |
| shift left command .....                | 388                |
| shift right command .....               | 388                |
| shiftwidth option.....                  | 395                |
| showmatch option .....                  | 395                |
| showmode command .....                  | 395                |
| slowopen command .....                  | 395                |
| source command .....                    | 381                |
| substitute command .....                | 381                |
| suspend command.....                    | 382                |
| tabstop option .....                    | 395                |
| tag command.....                        | 382                |
| taglength option .....                  | 395                |
| tags command.....                       | 396                |
| term command.....                       | 396                |
| terse command.....                      | 396                |
| unabbrev command.....                   | 383                |
| undo command.....                       | 383                |
| unmap command .....                     | 383                |
| version command.....                    | 383                |
| visual command .....                    | 384                |
| warn command.....                       | 396                |
| window command.....                     | 396                |
| wrapmargin option.....                  | 397                |
| wrapscan option.....                    | 397                |
| write command.....                      | 384                |
| write line number command .....         | 388                |
| writeany option .....                   | 397                |
| xit command.....                        | 385                |
| yank command .....                      | 385                |
| exec.....                               | 75, 674            |
| exec family .....                       | 28, 261, 659, 1058 |
| exec special built-in .....             | 75                 |
| Execution_Time .....                    | 110                |
| EXINIT .....                            | 353                |
| exit special built-in .....             | 77                 |
| exit status and errors.....             | 46                 |
| exit status for commands.....           | 46                 |
| expand .....                            | 424                |
| export special built-in .....           | 79                 |
| expr.....                               | 427                |
| matching expression.....                | 429                |
| string operand.....                     | 429                |
| expression argument .....               | 165                |
| expression list.....                    | 165                |
| EXPR_NEST_MAX.....                      | 18                 |
| extended regular expression .....       | 153, 161, 269      |
| .....452, 492, 536, 654, 707, 821, 1056 |                    |
| extension .....                         |                    |
| CX .....                                | 10                 |
| OH .....                                | 12                 |
| XSL.....                                | 16                 |
| false .....                             | 28, 48, 432        |
| fc .....                                | 28, 48, 434        |
| FD.....                                 | 10                 |
| fg .....                                | 28, 48, 440        |
| field splitting.....                    | 42                 |
| FIFO special files.....                 | 638                |
| file .....                              | 442                |
| file access permissions.....            | 4                  |
| file comparisons .....                  |                    |
| cmp .....                               | 251                |

|                               |       |                                   |             |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
|-------------------------------|-------|-----------------------------------|-------------|---------|-----|------------------|---|-----------------------------|-----|---------|--|-----------------|-----|----------|-----|-----------|-----|----------|-----|-----------|-----|----------------|-----|------------|------------|----------|-----|----------|------------|--------------|-----|-------------------------------|----|-----------|-----|------------------------------|---|-----------|-----|----------------------|-----|--|--|----------------------------|-----|
| comm .....                    | 254   | iconv .....                       | 503         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| diff.....                     | 317   | more .....                        | 640         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| uniq.....                     | 956   | nl .....                          | 665         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file contents .....           | 6     | paste .....                       | 684         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file conversion               |       | pax .....                         | 698         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| cut .....                     | 286   | pr.....                           | 733         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| dd .....                      | 300   | read.....                         | 813         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| expand .....                  | 424   | sed.....                          | 838         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| fold.....                     | 458   | tail.....                         | 894         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| head.....                     | 500   | tee.....                          | 901         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| join .....                    | 522   | tr.....                           | 923         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| od .....                      | 676   | uncompress .....                  | 948         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| paste .....                   | 684   | unexpand .....                    | 951         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| patch.....                    | 688   | zcat .....                        | 1076        |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| sort.....                     | 867   | find.....                         | <b>449</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| strings .....                 | 876   | fold.....                         | <b>458</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| tail.....                     | 894   | for loop.....                     | 52          |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| tr .....                      | 923   | fort77 .....                      | <b>461</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| tsort .....                   | 931   | external symbols.....             | 464         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| unexpand .....                | 951   | standard libraries.....           | 463         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| uniq.....                     | 956   | FR .....                          | <b>10</b>   |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| uudecode.....                 | 966   | FSC .....                         | <b>10</b>   |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| uuencode.....                 | 969   | function definition command ..... | 54          |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file creation.....            | 4     | function identifiers .....        | 198         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file descriptor.....          | 3, 43 | fuser .....                       | <b>467</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file mode creation mask ..... | 3     | g-file .....                      | 309         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file permission commands      |       | gencat .....                      | <b>470</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| chgrp .....                   | 233   | escape sequences .....            | 471         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| chmod .....                   | 236   | generated file .....              | 309         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| chown .....                   | 242   | get .....                         | <b>473</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| umask.....                    | 939   | getconf .....                     | <b>481</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file read .....               | 4     | getopts .....                     | 28, 48, 487 |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file removal .....            | 6     | global storage class .....        | 202         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file searching                |       | GNU make .....                    | 624         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| grep.....                     | 492   | grep .....                        | <b>492</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file time values .....        | 6     | grouping commands .....           | 52          |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file tree commands            |       | hash .....                        | <b>497</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| diff.....                     | 317   | head .....                        | <b>500</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| find.....                     | 449   | here-document .....               | 44          |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| ls .....                      | 568   | history command                   |             |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| mkdir.....                    | 635   | rmdir .....                       | 827         | fc..... | 434 | file write ..... | 4 | Hold Batch Job Request..... | 116 | filters |  | Hold_Types..... | 110 | asa..... | 141 | HOME..... | 370 | awk..... | 153 | hunk..... | 690 | compress ..... | 262 | iconv..... | <b>503</b> | dd ..... | 300 | id ..... | <b>506</b> | expand ..... | 424 | if conditional construct..... | 53 | fold..... | 458 | implementation-defined ..... | 1 | head..... | 500 | inference rule ..... | 607 |  |  | input field separator..... | 849 |
| rmdir .....                   | 827   | fc.....                           | 434         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| file write .....              | 4     | Hold Batch Job Request.....       | 116         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| filters                       |       | Hold_Types.....                   | 110         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| asa.....                      | 141   | HOME.....                         | 370         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| awk.....                      | 153   | hunk.....                         | 690         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| compress .....                | 262   | iconv.....                        | <b>503</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| dd .....                      | 300   | id .....                          | <b>506</b>  |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| expand .....                  | 424   | if conditional construct.....     | 53          |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| fold.....                     | 458   | implementation-defined .....      | 1           |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
| head.....                     | 500   | inference rule .....              | 607         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |
|                               |       | input field separator.....        | 849         |         |     |                  |   |                             |     |         |  |                 |     |          |     |           |     |          |     |           |     |                |     |            |            |          |     |          |            |              |     |                               |    |           |     |                              |   |           |     |                      |     |  |  |                            |     |

|                                       |                                      |
|---------------------------------------|--------------------------------------|
| input mode.....                       | 334                                  |
| IP6 .....                             | 11                                   |
| ipcrm .....                           | 510                                  |
| ipcs .....                            | 512                                  |
| I_ISVTX.....                          | 238                                  |
| jobs .....                            | 28, 48, 518                          |
| Job_Owner .....                       | 110                                  |
| join .....                            | 522                                  |
| Join_Path .....                       | 110                                  |
| Keep_Files .....                      | 110                                  |
| keyword-value pairs.....              | 123                                  |
| kill.....                             | 28, 48, 527                          |
| legacy.....                           | 1                                    |
| lex.....                              | 532                                  |
| actions.....                          | 538                                  |
| definitions .....                     | 534                                  |
| escape sequences.....                 | 537                                  |
| regular expressions .....             | 536                                  |
| rules.....                            | 535                                  |
| table sizes.....                      | 535                                  |
| user subroutines .....                | 536                                  |
| lex, translation table.....           | 542                                  |
| libraries                             |                                      |
| ar command.....                       | 134                                  |
| LIMIT .....                           | 17                                   |
| line counting.....                    | 1042                                 |
| LINE_MAX.....                         | 18, 154, 346-347, 354, 604, 846, 959 |
| link .....                            | 544                                  |
| lists.....                            | 50                                   |
| AND-OR.....                           | 50                                   |
| compound-list .....                   | 50                                   |
| ln .....                              | 546                                  |
| locale .....                          | 550                                  |
| localedef.....                        | 555                                  |
| Locate Batch Job Request.....         | 117                                  |
| locking file.....                     | 240                                  |
| logger .....                          | 559                                  |
| logname .....                         | 561                                  |
| lp .....                              | 563                                  |
| LR(1) grammars.....                   | 1074                                 |
| ls.....                               | 568                                  |
| m4 .....                              | 576                                  |
| macro processor.....                  | 576                                  |
| magic file .....                      | 447                                  |
| mailx.....                            | 583                                  |
| change current directory.....         | 594                                  |
| change folder.....                    | 595                                  |
| command escapes .....                 | 601                                  |
| commands.....                         | 593                                  |
| copy messages .....                   | 594                                  |
| declare aliases .....                 | 593                                  |
| declare alternatives .....            | 594                                  |
| delete aliases.....                   | 600                                  |
| delete messages .....                 | 594                                  |
| delete messages and display.....      | 595                                  |
| direct messages to mbox.....          | 597                                  |
| discard header fields.....            | 594                                  |
| display beginning of messages.....    | 600                                  |
| display current message number.....   | 601                                  |
| display header summary .....          | 596                                  |
| display list of folders .....         | 596                                  |
| display message.....                  | 598                                  |
| display message size.....             | 599                                  |
| echo a string.....                    | 595                                  |
| edit message .....                    | 595, 600                             |
| execute commands conditionally .....  | 597                                  |
| exit .....                            | 595                                  |
| follow up specified messages .....    | 596                                  |
| help .....                            | 596                                  |
| hold messages .....                   | 596                                  |
| internal variables .....              | 590                                  |
| invoke a shell.....                   | 599                                  |
| invoke shell command .....            | 601                                  |
| list available commands .....         | 597                                  |
| mail a message .....                  | 597                                  |
| null command .....                    | 601                                  |
| pipe message .....                    | 597                                  |
| process next specified message .....  | 597                                  |
| quit.....                             | 598                                  |
| read mailx commands from a file ..... | 599                                  |
| receive mode .....                    | 583                                  |
| reply to a message.....               | 598                                  |
| reply to a message list .....         | 598                                  |
| retain header fields.....             | 599                                  |
| save messages .....                   | 599                                  |
| scroll header display.....            | 601                                  |
| send mode.....                        | 583                                  |
| set variables .....                   | 599                                  |
| start-up .....                        | 590                                  |
| touch messages .....                  | 600                                  |
| undelete messages .....               | 600                                  |
| unset variables .....                 | 600                                  |
| write messages to a file .....        | 600                                  |
| Mail_Points .....                     | 111                                  |
| Mail_Users .....                      | 111                                  |
| make .....                            | 607                                  |
| default rules .....                   | 617                                  |
| inference rules .....                 | 614                                  |
| internal macros .....                 | 616                                  |
| libraries .....                       | 615                                  |
| macros.....                           | 613                                  |
| makefile execution .....              | 611                                  |

|                                          |            |                                                    |               |
|------------------------------------------|------------|----------------------------------------------------|---------------|
| makefile syntax .....                    | 610        | MX.....                                            | <b>12</b>     |
| target rules.....                        | 611        | NAME_MAX.....                                      | 138, 724      |
| make, GNU version .....                  | 624        | newgrp .....                                       | 28, 48, 657   |
| man.....                                 | <b>628</b> | NGROUPS_MAX .....                                  | 660           |
| mathematical functions.....              | 9          | nice.....                                          | <b>661</b>    |
| may .....                                | 2          | Ninth Edition UNIX .....                           | 205, 333, 742 |
| MC1 .....                                | <b>11</b>  | nl .....                                           | <b>665</b>    |
| MC2 .....                                | <b>11</b>  | nm .....                                           | <b>669</b>    |
| mesg .....                               | <b>632</b> | noclobber option.....                              | 697           |
| message catalog generation .....         | 470        | nohup .....                                        | <b>673</b>    |
| MF.....                                  | <b>11</b>  | non-printable .....                                | 346, 845, 900 |
| MIL-STD-1753 .....                       | 465        | OB .....                                           | <b>12</b>     |
| Minimum_Cpu_Interval.....                | 108        | object files.....                                  | 669           |
| mkdir.....                               | <b>635</b> | od .....                                           | <b>676</b>    |
| mkfifo .....                             | <b>638</b> | OF.....                                            | <b>12</b>     |
| ML.....                                  | <b>11</b>  | OH .....                                           | <b>12</b>     |
| MLR .....                                | <b>11</b>  | open file descriptors for reading and writing..... | 46            |
| Modify Batch Job Request .....           | 117        | open mode .....                                    | 353           |
| MON.....                                 | <b>12</b>  | option                                             |               |
| more.....                                | <b>640</b> | ADV.....                                           | 9             |
| discard and refresh.....                 | 646        | AIO .....                                          | 9             |
| display position .....                   | 649        | BAR.....                                           | 9             |
| examine new file.....                    | 648        | BE .....                                           | 10            |
| examine next file.....                   | 648        | CD .....                                           | 10            |
| examine previous file .....              | 648        | CPT .....                                          | 10            |
| go to beginning of file.....             | 646        | CS .....                                           | 10            |
| go to end-of-file.....                   | 646        | FD .....                                           | 10            |
| go to tag.....                           | 648        | FR .....                                           | 10            |
| help.....                                | 645        | FSC .....                                          | 10            |
| invoke editor .....                      | 648        | IP6 .....                                          | 11            |
| mark position .....                      | 647        | MC1 .....                                          | 11            |
| quit.....                                | 649        | MC2 .....                                          | 11            |
| refresh the screen.....                  | 646        | MF .....                                           | 11            |
| repeat search.....                       | 647        | ML .....                                           | 11            |
| repeat search in reverse.....            | 648        | MLR.....                                           | 11            |
| return to mark .....                     | 647        | MON .....                                          | 12            |
| return to previous position .....        | 647        | MPR.....                                           | 12            |
| scroll backward one half screenful ..... | 646        | MSG.....                                           | 12            |
| scroll backward one line .....           | 645        | MX .....                                           | 12            |
| scroll backward one screenful .....      | 645        | PIO .....                                          | 13            |
| scroll forward one half screenful .....  | 646        | PS .....                                           | 13            |
| scroll forward one line .....            | 645        | RS .....                                           | 13            |
| scroll forward one screenful .....       | 645        | RTS .....                                          | 13            |
| search backward for pattern .....        | 647        | SD .....                                           | 13            |
| search forward for pattern .....         | 647        | SEM .....                                          | 13            |
| skip forward one line.....               | 646        | SHM .....                                          | 13            |
| motion command .....                     | 853        | SIO .....                                          | 13            |
| Move Batch Job Request .....             | 118        | SPI .....                                          | 14            |
| MPR .....                                | <b>12</b>  | SPN .....                                          | 14            |
| MSG .....                                | <b>12</b>  | SS.....                                            | 14            |
| mv .....                                 | <b>652</b> | TCT .....                                          | 14            |

|                                          |                    |                                    |             |
|------------------------------------------|--------------------|------------------------------------|-------------|
| TEF                                      | 14                 | cpio filename                      | 720         |
| THR                                      | 14                 | cpio header                        | 718         |
| TMO                                      | 14                 | cpio interchange format            | 718         |
| TMR                                      | 15                 | cpio special entries               | 720         |
| TPI                                      | 15                 | extended header                    | 711         |
| TPP                                      | 15                 | extended header file times         | 714         |
| TPS                                      | 15                 | extended header keyword precedence | 714         |
| TRC                                      | 15                 | list mode format specifications    | 706         |
| TRI                                      | 15                 | ustar format                       | 714         |
| TRL                                      | 15                 | ustar interchange format           | 714         |
| TSA                                      | 15                 | PIO                                | 13          |
| TSF                                      | 16                 | pipelines                          | 49          |
| TSH                                      | 16                 | portable character set             | 613         |
| TSP                                      | 16                 | positional parameters              | 33          |
| TSS                                      | 16                 | POSIX2_BC_BASE_MAX                 | 17-18       |
| TYM                                      | 16                 | POSIX2_BC_DIM_MAX                  | 17-18       |
| UP                                       | 16                 | POSIX2_BC_SCALE_MAX                | 17-18       |
| XSR                                      | 17                 | POSIX2_BC_STRING_MAX               | 17-18       |
| OR lists                                 | 51                 | POSIX2_COLL_WEIGHTS_MAX            | 17-18       |
| ordinary identifiers                     | 198                | POSIX2_EXPR_NEST_MAX               | 17, 19      |
| Output_Path                              | 111                | POSIX2_LINE_MAX                    | 17, 19      |
| paginators                               |                    | POSIX2_RE_DUP_MAX                  | 17, 19      |
| more                                     | 640                | POSIX2_SYMLINKS                    | 19          |
| parameter expansion                      | 37                 | POSIX2_VERSION                     | 17          |
| parameters and variables                 | 33                 | pr                                 | 733         |
| paste                                    | <b>684</b>         | print-related commands             |             |
| patch                                    | <b>688</b>         | fold                               | 458         |
| filename determination                   | 691                | lp                                 | 563         |
| patch application                        | 691                | pr                                 | 733         |
| patch file format                        | 690                | printf                             | 738         |
| patchchk                                 | <b>694</b>         | Priority                           | 112         |
| pathname expansion                       | 42                 | privileges                         | 633, 663    |
| pathname manipulation                    |                    | process attributes                 | 3           |
| basename                                 | 187                | process group ID                   | 3           |
| dirname                                  | 324                | process ID                         | 3           |
| patchchk                                 | 694                | process status report              | 748         |
| pathname resolution                      | 7                  | prs                                | 743         |
| PATH_MAX                                 | 18, 730, 823       | PS                                 | 13          |
| pattern matching                         | 449, 707, 963, 979 | ps                                 | 748         |
| definition                               | 62                 | public locale                      | 550         |
| in case statements                       | 53                 | pwd                                | 28, 48, 755 |
| in shell variables                       | 39                 | qalter                             | 757         |
| pattern matching notation                | 62, 455, 723       | qdel                               | 766         |
| pattern scanning and processing language |                    | qhold                              | 769         |
| at                                       | 153                | qmove                              | 772         |
| patterns matching a single character     | 62                 | qmsg                               | 775         |
| patterns matching multiple characters    | 63                 | qrerun                             | 778         |
| patterns used for filename expansion     | 63                 | qrls                               | 780         |
| pax                                      | <b>698</b>         | qselect                            | 783         |
| archive character set encoding/decoding  | 729                | qsig                               | 792         |
| cpio file data                           | 720                | qstat                              | 795         |

|                                   |                                      |                                                    |             |
|-----------------------------------|--------------------------------------|----------------------------------------------------|-------------|
| qsub .....                        | <b>800</b>                           | addresses.....                                     | 840         |
| Queue Batch Job Request.....      | 118                                  | editing commands.....                              | 840         |
| quote removal .....               | 42                                   | regular expressions .....                          | 840         |
| quoting.....                      | 30                                   | Select Batch Jobs Request.....                     | 120         |
| read.....                         | 28, 48, 813                          | SEM.....                                           | <b>13</b>   |
| readonly special built-in.....    | <b>81</b>                            | sequential lists .....                             | 51          |
| real group ID.....                | 3                                    | Server Shutdown Request .....                      | 120         |
| real user ID.....                 | 3                                    | Server Status Request .....                        | 121         |
| redirecting input .....           | 44                                   | session membership.....                            | 3           |
| redirecting output.....           | 44                                   | set special built-in.....                          | <b>85</b>   |
| redirection .....                 | 43                                   | set-group-ID .....                                 | 3, 271      |
| regular expressions .....         | 161, 269, 335, 359                   | set-user-ID .....                                  | 3, 241, 271 |
| .....                             | 389, 429, 452, 492, 536, 644, 654    | set-user-ID scripts .....                          | 861         |
| .....                             | 665, 704, 821, 840, 1001, 1004, 1056 | sh.....                                            | <b>847</b>  |
| related to shell patterns.....    | 62                                   | command history list.....                          | 851         |
| relational database operator..... | 522                                  | command line editing.....                          | 851         |
| Release Batch Job Request .....   | 119                                  | vi line editing command mode .....                 | 853         |
| remove directories.....           | 827                                  | vi line editing insert mode.....                   | 852         |
| remove files.....                 | 820                                  | vi-mode command line editing.....                  | 852         |
| renice .....                      | <b>816</b>                           | shall.....                                         | 2           |
| requested batch services .....    | 115                                  | shell command language .....                       | 29          |
| Rerun Batch Job Request.....      | 120                                  | alias substitution .....                           | 32          |
| Rerunable .....                   | 112                                  | appending redirected output.....                   | 44          |
| reserved words.....               | 33                                   | arithmetic expansion .....                         | 41          |
| Resource_List .....               | 112                                  | command substitution .....                         | 40          |
| return special built-in .....     | <b>83</b>                            | compound commands.....                             | 52          |
| RE_DUP_MAX.....                   | 18                                   | consequences of shell errors.....                  | 46          |
| rm.....                           | <b>820</b>                           | double-quotes.....                                 | 30          |
| rmdel .....                       | <b>825</b>                           | duplicating an input file descriptor .....         | 45          |
| rmdir.....                        | <b>827</b>                           | duplicating an output file descriptor .....        | 45          |
| root directory.....               | 3                                    | escape character (backslash) .....                 | 30          |
| RS .....                          | <b>13</b>                            | exit status and errors .....                       | 46          |
| RTS.....                          | <b>13</b>                            | exit status for commands.....                      | 46          |
| sact.....                         | <b>830</b>                           | field splitting .....                              | 42          |
| saved set-group-ID.....           | 3                                    | function definition command .....                  | 54          |
| saved set-user-ID .....           | 3                                    | grammar .....                                      | 55          |
| sccs.....                         | <b>833</b>                           | here-document .....                                | 44          |
| SCCS commands                     |                                      | introduction.....                                  | 29          |
| admin .....                       | 126                                  | lists.....                                         | 50          |
| delta.....                        | 309                                  | open file descriptors for reading and writing..... | 46          |
| get .....                         | 473                                  | parameter expansion .....                          | 37          |
| prs .....                         | 743                                  | parameters and variables.....                      | 33          |
| rmdel.....                        | 825                                  | pathname expansion.....                            | 42          |
| sact.....                         | 830                                  | pattern matching notation .....                    | 62          |
| sccs.....                         | 833                                  | patterns matching a single character .....         | 62          |
| unget .....                       | 954                                  | patterns matching multiple characters .....        | 63          |
| val .....                         | 981                                  | patterns used for filename expansion .....         | 63          |
| what .....                        | 1045                                 | pipelines .....                                    | 49          |
| SD.....                           | <b>13</b>                            | positional parameters .....                        | 33          |
| search pattern.....               | 281                                  | quote removal .....                                | 42          |
| sed.....                          | <b>838</b>                           | quoting .....                                      | 30          |

|                                             |                                   |
|---------------------------------------------|-----------------------------------|
| redirecting input .....                     | 44                                |
| redirecting output .....                    | 44                                |
| redirection.....                            | 43                                |
| reserved words .....                        | 33                                |
| shell commands .....                        | 47                                |
| shell execution environment.....            | 61                                |
| shell grammar lexical conventions .....     | 55                                |
| shell grammar rules .....                   | 56                                |
| shell variables.....                        | 34                                |
| signals and error handling.....             | 61                                |
| simple commands .....                       | 47                                |
| single-quotes .....                         | 30                                |
| special built-in utilities.....             | 64                                |
| special parameters.....                     | 34                                |
| tilde expansion.....                        | 37                                |
| token recognition.....                      | 31                                |
| word expansions .....                       | 36                                |
| shell commands .....                        | 47                                |
| shell execution environment.....            | 61, 132, 815, 941                 |
| shell grammar.....                          | 55                                |
| shell grammar lexical conventions .....     | 55                                |
| shell grammar rules .....                   | 56                                |
| shell introduction.....                     | 29                                |
| shell variables.....                        | 34                                |
| Shell_Path_List.....                        | 112                               |
| shift special built-in.....                 | 91                                |
| SHM.....                                    | 13                                |
| should.....                                 | 2                                 |
| SIGCONT.....                                | 356                               |
| SIGHUP .....                                | 335, 356, 984, 1025               |
| SIGINT .....                                | 311, 335, 355, 1037               |
| Signal Batch Job Request.....               | 121                               |
| signal processes .....                      | 527                               |
| signals and error handling.....             | 61                                |
| SIGQUIT .....                               | 335                               |
| SIGTERM.....                                | 356                               |
| simple commands .....                       | 47                                |
| single-quotes .....                         | 30                                |
| SIO .....                                   | 13                                |
| sleep.....                                  | 864                               |
| SLR(1) grammars.....                        | 1074                              |
| sort.....                                   | 867                               |
| special built-in.....                       | 261, 663, 675, 756, 861, 914, 930 |
| special built-in utilities.....             | 64                                |
| break.....                                  | 65, 69                            |
| characteristics.....                        | 64                                |
| colon.....                                  | 67                                |
| dot.....                                    | 71                                |
| eval .....                                  | 73                                |
| exec.....                                   | 75                                |
| exit .....                                  | 77                                |
| export .....                                | 79                                |
| readonly.....                               | 81                                |
| return.....                                 | 83                                |
| set .....                                   | 85                                |
| shift.....                                  | 91                                |
| times .....                                 | 93                                |
| trap .....                                  | 95                                |
| unset .....                                 | 98                                |
| special parameters.....                     | 34                                |
| special targets.....                        | 612                               |
| SPI.....                                    | 14                                |
| split .....                                 | 873                               |
| split files                                 |                                   |
| csplit .....                                | 277                               |
| split .....                                 | 873                               |
| SPN.....                                    | 14                                |
| spoofing .....                              | 82                                |
| SS.....                                     | 14                                |
| standard error.....                         | 43                                |
| standard input.....                         | 43                                |
| standard output .....                       | 43                                |
| strings.....                                | 876                               |
| strip.....                                  | 879                               |
| stty .....                                  | 881                               |
| combination modes .....                     | 886                               |
| control modes.....                          | 881                               |
| input modes.....                            | 882                               |
| local modes.....                            | 884                               |
| output modes .....                          | 883                               |
| special control character assignments ..... | 885                               |
| st_gid.....                                 | 139                               |
| st_mode .....                               | 139                               |
| st_mtime .....                              | 139                               |
| st_size .....                               | 139                               |
| st_uid.....                                 | 139                               |
| superuser .....                             | 438, 574, 722                     |
| supplementary group IDs .....               | 3                                 |
| system configuration values .....           | 481                               |
| system name .....                           | 945                               |
| tabs.....                                   | 890                               |
| tag file creation.....                      | 281                               |
| tail .....                                  | 894                               |
| talk .....                                  | 897                               |
| tar format.....                             | 714                               |
| target queue .....                          | 118                               |
| target rule .....                           | 607                               |
| TCT .....                                   | 14                                |
| tee .....                                   | 901                               |
| TEF.....                                    | 14                                |
| terminal characteristics .....              |                                   |
| stty .....                                  | 881                               |

|                              |             |                               |                    |
|------------------------------|-------------|-------------------------------|--------------------|
| tabs .....                   | 890         | user identity                 |                    |
| tput .....                   | 920         | id .....                      | 506                |
| tty .....                    | 933         | logname .....                 | 561                |
| terminate processes.....     | 527         | newgrp .....                  | 657                |
| terminology.....             | 1           | who .....                     | 1047               |
| test.....                    | 904         | User_List.....                | 113                |
| THR .....                    | 14          | ustar format .....            | 714                |
| tilde expansion.....         | 37          | utility option parsing.....   | 487                |
| time.....                    | 912         | uucp .....                    | 962                |
| times special built-in.....  | 93          | uudecode .....                | 966                |
| TMO .....                    | 14          | uuencode .....                | 969                |
| TMPDIR.....                  | 702         | uustat .....                  | 974                |
| TMR .....                    | 15          | uux .....                     | 977                |
| token recognition.....       | 31          | val .....                     | 981                |
| touch.....                   | 916         | Variable_List.....            | 113                |
| TPI.....                     | 15          | vi .....                      | 984                |
| TPP.....                     | 15          | <ESC> .....                   | 1024               |
| TPS.....                     | 15          | append .....                  | 1005               |
| tput .....                   | 920         | change .....                  | 1006               |
| tr.....                      | 923         | change to end-of-line.....    | 1006               |
| Track Batch Job Request..... | 121         | clear and redisplay .....     | 992                |
| trap special built-in .....  | 95          | command descriptions.....     | 985                |
| TRC.....                     | 15          | control-D .....               | 1021               |
| TRI .....                    | 15          | control-H .....               | 1021               |
| TRL .....                    | 15          | control-T .....               | 1023               |
| trojan horse .....           | 574         | control-U .....               | 1023               |
| true .....                   | 28, 48, 929 | control-V .....               | 1023               |
| TSA .....                    | 15          | current and line above.....   | 999                |
| TSF .....                    | 16          | delete .....                  | 1007               |
| TSH.....                     | 16          | delete character.....         | 1016-1017          |
| tsort.....                   | 931         | delete to end-of-line.....    | 1007               |
| TSP .....                    | 16          | display information .....     | 991                |
| TSS .....                    | 16          | edit the alternate file ..... | 993                |
| tty .....                    | 933         | enter ex mode.....            | 1013               |
| TYM.....                     | 16          | execute .....                 | 1004               |
| type .....                   | 935         | execute an ex command.....    | 1003               |
| ulimit .....                 | 937         | exit .....                    | 1019               |
| umask .....                  | 28, 48, 939 | find character .....          | 1008               |
| unalias .....                | 28, 48, 943 | find regular expression ..... | 1001               |
| uname .....                  | 945         | Initialization .....          | 985                |
| unary primaries.....         | 905         | input mode commands .....     | 1019               |
| uncompress.....              | 948         | insert .....                  | 1009-1010          |
| undefined.....               | 2           | insert empty line .....       | 1011-1012          |
| unexpand.....                | 951         | join .....                    | 1010               |
| unget.....                   | 954         | mark position .....           | 1010               |
| uniq.....                    | 956         | move back.....                | 999, 1005          |
| unlink .....                 | 960         | move cursor .....             | 991, 994-995, 1014 |
| unset special built-in.....  | 98          | move down.....                | 992                |
| unspecified.....             | 2           | move forward.....             | 999                |
| until loop .....             | 54          | move to bigword.....          | 1008, 1016         |
| UP.....                      | 16          | move to bottom of screen..... | 1010               |

|                                         |              |
|-----------------------------------------|--------------|
| move to first character in line .....   | 1002         |
| move to first non-<blank> .....         | 998          |
| move to line .....                      | 1009         |
| move to matching character .....        | 995          |
| move to middle of screen .....          | 1011         |
| move to next section .....              | 998          |
| move to specific column .....           | 1000         |
| move to top of screen .....             | 1009         |
| move to word .....                      | 1008, 1015   |
| move up .....                           | 992          |
| newline .....                           | 1022         |
| nul .....                               | 1021, 1024   |
| page backwards .....                    | 990          |
| page forward .....                      | 991          |
| put from buffer .....                   | 1012         |
| redraw screen .....                     | 993          |
| redraw window .....                     | 1018         |
| regular expression .....                | 1004         |
| repeat .....                            | 1000         |
| repeat find .....                       | 1003, 1011   |
| repeat substitution .....               | 996          |
| replace character .....                 | 1013-1014    |
| replace text with command .....         | 994          |
| return to previous context .....        | 996-997      |
| return to previous section .....        | 997          |
| reverse case .....                      | 1004         |
| reverse find character .....            | 1000         |
| scroll backward .....                   | 993          |
| scroll backward by line .....           | 993          |
| scroll forward .....                    | 990          |
| scroll forward by line .....            | 990          |
| search for tagstring .....              | 994          |
| shift left .....                        | 1003         |
| shift right .....                       | 1003         |
| substitute character .....              | 1014         |
| substitute lines .....                  | 1014         |
| terminate command or input mode .....   | 993          |
| undo .....                              | 1015         |
| undo current line .....                 | 1015         |
| yank .....                              | 1017         |
| yank current line .....                 | 1017         |
| visual mode .....                       | 353          |
| wait .....                              | 28, 48, 1038 |
| warning                                 |              |
| OB .....                                | 12           |
| OF .....                                | 12           |
| wc .....                                | <b>1042</b>  |
| what .....                              | <b>1045</b>  |
| while loop .....                        | 54           |
| who .....                               | <b>1047</b>  |
| word counting .....                     | 1042         |
| word expansions .....                   | 36           |
| write .....                             | <b>1051</b>  |
| xargs .....                             | <b>1054</b>  |
| XSI .....                               | <b>16</b>    |
| XSR .....                               | <b>17</b>    |
| yacc .....                              | <b>1060</b>  |
| algorithms .....                        | 1071         |
| code file .....                         | 1062         |
| completing the program .....            | 1071         |
| conflicts .....                         | 1069         |
| debugging the parser .....              | 1071         |
| declarations section .....              | 1063         |
| description file .....                  | 1062         |
| error handling .....                    | 1069         |
| grammar rules .....                     | 1065         |
| header file .....                       | 1062         |
| input grammar .....                     | 1067         |
| input language .....                    | 1062         |
| interface to the lexical analyzer ..... | 1070         |
| lexical structure of the grammar .....  | 1063         |
| library .....                           | 1071         |
| limits .....                            | 1072         |
| programs section .....                  | 1066         |
| zcat .....                              | <b>1076</b>  |