IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

### E.3.13 Pattern Matching Notation

### E.3.14 Special Built-in Utilities

## E.4 Execution Environment Utilities

*Editor's Note: The text of the Rationale for this section has been temporarily located in Section 4, adjacent to the text it is explaining. The text will return to this annex after the completion of balloting. Notations regarding utilities probably included in the UPE have been updated, without diff marks, based on the current working draft of 1003.2a.*

Many utilities were evaluated by the working group; more utilities were excluded from the standard than included. The following list contains many common UNIX system utilities that were not included as Execution Environment Utilities or in one of the Software Development Environment groups. It is logistically difficult for this Rationale to correctly distribute the reasons for not including a utility among the various utility environment sections. Therefore, this section covers the reasons for all utilities not included in Sections 4 and 6 and Annexes A and C.

The working group started its deliberations with a recommended list of utilities provided by the X/Open group of companies. This list was a subset of the utilities in the *X/Open Portability Guide*, *Issue II*, so it was very closely related to System V. The list had already been purged of purely administrative utilities, such as those found in System V's Administered System Extension. Then, the working group applied its scope as a filter and substantially pruned the remaining list as well.

The following list of ''rejected'' utilities is limited by its historical roots; since the selected utilities emerged from primarily a System V base, this list does not include sometimes familiar entries from BSD. The working group received substantial input from representatives of the University of California at Berkeley and from companies that are firmly allied with BSD versions of the UNIX system, enough so that some BSD-derived utilities are included in the standard. However, this Rationale is now limited to a discussion of only those utilities actively or indirectly evaluated by the working group, rather than the list of all known UNIX utilities from all its variants. This list will most likely be augmented during the balloting process as balloters request specific rationales for their favorite commands.

In the list, the notation [*POSIX .2a*] is used to identify utilities that are being evaluated for inclusion in the forthcoming User Portability Extension to this standard. Similarly, [*POSIX .7*] is used for those that may be appropriate for the working group evaluating system administration and [*POSIX .Net*] for networking standards.

`adb` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

tool. Furthermore, many useful aspects of `adb` are very hardware-specific.

`admin` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`as` Assemblers are hardware-specific and are included implicitly as part of the compilers in the standard.

`at` The `at` and `cron` family of utilities were omitted because portable applications could not rely on their behavior. [*POSIX .2a* ]

`banner` The only known use of this command is as part of the LP printer header pages. It was decided that the format of the header is implementation defined, so this utility is superfluous to application portability.

`batch` The `at` and `cron` family of utilities were omitted because portable applications could not rely on their behavior. [*POSIX .2a* ]

`cal` This calendar printing program is not useful to portable applications.

`calendar` This reminder service program is not useful to portable applications.

`cancel` The LP (line printer spooling) system specified is the most basic possible and did not need this level of application control. [*POSIX .7* ]

`cflow` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool.

`chroot` This is primarily of administrative use, requiring super-user privileges. [*POSIX .7* ]

`col` No utilities defined in this standard produce output requiring such a filter. The `nroff` text formatter is present on many historical systems and will continue to remain as an extension; `col` is expected to be shipped by all the systems that ship `nroff`.

`cpio` This has been replaced by `pax`, for reasons explained in its own Rationale.

`cpp` Can be subsumed by `c89`.

`crontab` The `at` and `cron` family of utilities were omitted because portable applications could not rely on their behavior. [*POSIX .2a* ]

`csplit` This utility's functionality can sometimes be provided by the `dd` or `sed` utilities (i.e., although these utilities cannot easily provide all of `csplit`'s features in one package, they can frequently be

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

used for the type of task that `csplit` is being used for). [*POSIX .2a* ]

`cu` Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .Net* ]

`cxref` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool.

`dc` This utility's functionality can be provided by the `bc` utility; `bc` was selected because it was easier to use and had superior functionality. Although the historical versions of `bc` are implemented using `dc` as a base, this standard prescribes the interface and not the underlying mechanism used to implement it.

`delta` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`df` As the standard does not address the concept or nature of file systems, this command could not be specified in a manner useful to portable applications. [*POSIX .2a* ]

`dircmp` Although a useful concept, the traditional output of this directory comparison program is not suitable for processing in applications programs. Also, the `diff -r` command gives equivalent functionality.

`dis` Disassemblers are hardware-specific.

`du` Because of differences between systems in measuring disk usage, this utility could not be used reliably by a portable application. [*POSIX .2a* ]

`egrep` Marked obsolescent and replaced by the new version of `grep`.

`ex` This is typically a link to the `vi` terminal-oriented editor—not useful from shell scripts or typical application programs. The nonterminal oriented facilities of `ex` are provided by `ed`. [*POSIX .2a* ]

`fgrep` Marked obsolescent and replaced by the new version of `grep`.

`file` Determining the type of file is generally accomplished with `test` or `find`. The added information available with `file` is of little use to a portable application, particularly since there is considerable variation in its output contents. [*POSIX .2a* ]

`get` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

`ld` Is subsumed by `c89`.

`line` The functionality of `line` can be provided with `read`.

`lint` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool.

`login` Terminal oriented—not useful from shell scripts or typical application programs.

`lorder` This utility is an aid in creating an implementation-specific detail of object libraries that the working group did not feel required standardization.

`lpstat` The LP system specified is the most basic possible and did not need this level of application control. [*POSIX .7*]

`m4` The working group did not find that this macro processor had sufficiently wide usage for standardization.

`mail` This utility was omitted in favor of `mailx`, because there was a considerable functionality overlap between the two. The mail-sending aspects of `mailx` are covered in this standard, the mail-reading in the UPE. [*POSIX .2a*]

`mesg` Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .2a*]

`mknod` This was omitted in favor of `mkfifo`, as `mknod` has too many implementation-defined functions. [*POSIX .7*]

`newgrp` Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .2a*]

`news` Terminal oriented—not useful from shell scripts or typical application programs.

`nice` Due to historical variations in usage, and in the lack of underlying support from possible POSIX.1 {8}base systems, this cannot be used by applications to achieve reliable results. [*POSIX .2a*]

`nl` The useful functionality of `nl` can be provided with `pr`.

`nm` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool. [*POSIX .2a*]

`pack` The working group found little interest in a portable data compression program (and there are others that are probably more widely used anyway).

`passwd` Terminal oriented—not useful from shell scripts or typical application programs. (There was also sentiment to avoid security-

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

related utilities until requirements of 1003.6 are known.)

`pcat` The working group found little interest in a portable data compression program (and there are others that are probably more widely used anyway).

`pg` Terminal oriented—not useful from shell scripts or typical application programs.

`prof` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool.

`prs` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`ps` This utility has historically been difficult to specify portably due to the many implementation-defined aspects of processes. Furthermore, a portable application can rarely rely on information about what other processes are doing, as security mechanisms may prevent it. A process requiring one of its children's process IDs (such as for use with the `kill` command) will have to record the IDs at the time of creation. [*POSIX .2a* ]

`red` Restricted editor. This was not considered by the working group because it never provided the level of security restriction required.

`rmdel` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`rsh` Restricted shell. This was not considered by the working group because it does not provide the level of security restriction that is implied by historical documentation.

`sact` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`sdb` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool. Furthermore, some useful aspects of `sdb` are very hardware-specific.

`sdiff` The ''side-by-side `diff`'' utility from System V was omitted because it is used infrequently, and even less so by portable applications. Despite being in System V, it is not in the *SVID* or *XPG*.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

**shar** Utilities with this type of functionality (''shell-based archivers'') are in wide use, despite not being included in System V or BSD systems. However, the working group felt this sort of program was more widely used by human users than portable applications.

**shl** Terminal oriented—not useful from shell scripts or typical application programs. The job control aspects of the Shell Command Language are generally more useful and are being evaluated for the UPE.

**size** The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This utility is primarily a debugging tool.

**spell** Not useful from shell scripts or typical application programs.

**split** The functionality can sometimes be provided by the **dd**, **sed**, or (for some uses) **xargs** utilities (i.e., although these utilities cannot easily provide all of **split**'s features in one package, they can sometimes be used for the type of task that **split** is being used for). [*POSIX .2a* ]

**strings** This is normally used by human users during debugging, rather than by applications. [*POSIX .2a* ]

**su** Not useful from shell scripts or typical application programs. (There was also sentiment to avoid security-related utilities until requirements of POSIX. 6 are known.)

**sum** This utility was renamed **cksum**.

**tabs** Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .2a* ]

**time** Not necessary for portable applications. It is frequently used by human users in debugging or for informal benchmarks. It is doubtful whether any standardized definitions of the output could be agreed upon.

**tsort** This utility is an aid in creating an implementation-specific detail of object libraries that the working group did not feel required standardization.

**unget** The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

**unpack** The working group found little interest in a portable data compression program (and there are others that are probably more widely used anyway).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

`uucp`
`uulog`
`uupick`
`uustat`
`uuto` The UUCP utilities and their protocol description were removed from an early draft because responsibility for them was officially requested by the POSIX group developing networking interfaces.

`val` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`vi` Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .2a* ]

`wall` Terminal oriented—not useful from shell scripts or typical application programs. It is generally used by system administrators, as well. [*POSIX .7* ]

`what` The intent of the various software development utilities was to assist in the installation (rather than the actual development and debugging) of applications. This SCCS utility is primarily a development tool.

`who` The ability to determine other users on the system was felt to be at risk in a trusted implementation, so its use could not be considered by a portable application. [*POSIX .2a* ]

`write` Terminal oriented—not useful from shell scripts or typical application programs. [*POSIX .2a* ]

### E.4.1 `awk` — Pattern scanning and processing language

### E.4.2 `basename` — Return nondirectory portion of pathname

### E.4.3 `bc` — Arbitrary-precision arithmetic language

### E.4.4 `cat` — Concatenate and print files

### E.4.5 `cd` — Change working directory

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

333 **E.4.6 `chgrp` — Change file group ownership**

334 **E.4.7 `chmod` — Change file modes**

335 **E.4.8 `chown` — Change file ownership**

336 **E.4.9 `cksum` — Write file checksums and block counts**

337 **E.4.10 `cmp` — Compare two files**

338 **E.4.11 `comm` — Select or reject lines common to two files**

339 **E.4.12 `command` — Select or reject lines common to two files**

340 **E.4.13 `cp` — Copy files**

341 **E.4.14 `cut` — Cut out selected fields of each line of a file**

342 **E.4.15 `date` — Write the date and time**

343 **E.4.16 `dd` — Convert and copy a file**

344 **E.4.17 `diff` — Compare two files**

345 **E.4.18 `dirname` — Return directory portion of pathname**

346 **E.4.19 `echo` — Write arguments to standard output**

347 **E.4.20 `ed` — Edit text**

348 **E.4.21 `env` — Set environment for command invocation**

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### E.4.22 `expr` — Evaluate arguments as an expression

### E.4.23 `false` — Return false value

### E.4.24 `find` — Find files

### E.4.25 `fold` — Filter for folding lines

### E.4.26 `getconf` — Get configuration values

### E.4.27 `getopts` — Parse utility options

### E.4.28 `grep` — File pattern searcher

### E.4.29 `head` — Copy the first part of files

### E.4.30 `id` — Return user identity

### E.4.31 `join` — Relational database operator

### E.4.32 `kill` — Terminate or signal processes

### E.4.33 `ln` — Link files

### E.4.34 `locale` — Get locale-specific information

### E.4.35 `localedef` — Define locale environment

### E.4.36 `logger` — Log messages

### E.4.37 `logname` — Return user's login name

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

365 **E.4.38 `lp` — Send files to a printer**

366 **E.4.39 `ls` — List directory contents**

367 **E.4.40 `mailx` — Process messages**

368 **E.4.41 `mkdir` — Make directories**

369 **E.4.42 `mkfifo` — Make FIFO special files**

370 **E.4.43 `mv` — Move files**

371 **E.4.44 `nohup` — Invoke a utility immune to hangups**

372 **E.4.45 `od` — Dump files in various formats**

373 **E.4.46 `paste` — Merge corresponding or subsequent lines of files**

374 **E.4.47 `pathchk` — Check pathnames**

375 **E.4.48 `pax` — Portable archive interchange**

376 **E.4.49 `pr` — Print files**

377 **E.4.50 `printf` — Write formatted output**

378 **E.4.51 `pwd` — Return working directory name**

379 **E.4.52 `read` — Read a line from standard input**

380 **E.4.53 `rm` — Remove directory entries**

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

381 **E.4.54 `rmdir` — Remove directories**

382 **E.4.55 `sed` — Stream editor**

383 **E.4.56 `sh` — Shell, the standard command language interpreter**

384 **E.4.57 `sleep` — Suspend execution for an interval**

385 **E.4.58 `sort` — Sort, merge, or sequence check text files**

386 **E.4.59 `stty` — Set the options for a terminal**

387 **E.4.60 `tail` — Copy the last part of a file**

388 **E.4.61 `tee` — Duplicate standard input**

389 **E.4.62 `test` — Evaluate expression**

390 **E.4.63 `touch` — Change file access and modification times**

391 **E.4.64 `tr` — Translate characters**

392 **E.4.65 `true` — Return true value**

393 **E.4.66 `tty` — Return user's terminal name**

394 **E.4.67 `umask` — Get or set the file mode creation mask**

395 **E.4.68 `uname` — Return system name**

396 **E.4.69 `uniq` — Report or filter out repeated lines in a file**

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### E.4.70 `wait` — Await process completion

### E.4.71 `wc` — Word, line, and byte count

### E.4.72 `xargs` — Construct argument list(s) and invoke utility

## E.5 User Portability Utilities Option

*Editor's Note: This section is unused in this revision of the standard.*

## E.6 Software Development Utilities Option

*Editor's Note: The text of the Rationale for this section has been temporarily located in Section 6, adjacent to the text it is explaining. The text will return to this annex after the completion of balloting.*

This is the first of the optional utility environments. The working group decided there were two basic classes of systems to be supported: general application execution and software development. The first is widely used and is the primary reason for the development of this standard. The second, however, represents only a (small?) subset of the first; the users are generally only those who are developing or installing C or FORTRAN applications.

Therefore, all the development environments are optional, giving users the option of specifying a smaller, (presumably) less expensive system. There are three separate optional environments, so that C-only or FORTRAN -only users do not have to specify unneeded components. As further languages are supported by this standard, their environments will also be optional.

An implementation must provide all three of these utilities to claim conformance to this section.

See section **E.4** for a discussion of utilities excluded from this group.

### E.6.1 `ar` — Create and maintain library archives

### E.6.2 `make` — Maintain, update, and regenerate groups of programs

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.