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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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]

9620 #### 4.46.6.2 Standard Error

9621 Used only for diagnostic messages.

9622 #### 4.46.6.3 Output Files

9623 None.

9624 ### 4.46.7 Extended Description

9625 None.

9626 ### 4.46.8 Exit Status

9627 The `paste` utility shall exit with one of the following values:

9628 0 Successful completion.

9629 >0 An error occurred.

9630 ### 4.46.9 Consequences of Errors

9631 If one or more input files cannot be opened when the `-s` option is not specified, a
9632 diagnostic message shall be written to standard error, but no output shall be writ-
9633 ten to standard output. If the `-s` option is specified, the `paste` utility shall pro-
9634 vide the default behavior described in 2.11.9.

9635 ### 4.46.10 Rationale. *(This subclause is not a part of P1003.2)*

9636 **Examples, Usage**

9637 When the escape sequences of the *list* option-argument are used in a shell script,
9638 they must be quoted; otherwise, the shell treats the `\` as a special character.

9639 Write out a directory in four columns:

9640 ```
ls | paste - - - -
```

9641 Combine pairs of lines from a file into single lines:

9642 ```
paste -s -d "\t\n" file
```

9643 Portable applications should only use the specific backslash escaped delimiters
9644 presented in this standard. Historical implementations treat `\x`, where `x` is not
9645 in this list, as `x`, but future implementations are free to expand this list to recog-
9646 nize other common escapes similar to those accepted by `printf` and other stan-
9647 dard utilities.

9648 Most of the standard utilities work on text files. The `cut` utility can be used to
9649 turn files with arbitrary line lengths into a set of text files containing the same

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

9650 data. The `paste` utility can be used to create (or recreate) files with arbitrary
9651 line lengths. For example, if `file` contains long lines:

```
9652      cut -b 1-500 -n file > file1
9653      cut -b 501- -n file > file2
```

9654 creates `file1` (a text file) with lines no longer than 500 bytes (plus the <new-
9655 line> character) and `file2` that contains the remainder of the data from `file`.
9656 (Note that `file2` will not be a text file if there are lines in file that are longer
9657 than 500 + {LINE _MAX} bytes.) The original file can be recreated from `file1` and
9658 `file2` using the command:

```
9659      paste -d "\0" file1 file2 > file
```

9660 The commands 2

```
9661      paste -d "\0" . . .                                    2
9662      paste -d "" . . .                                      2
```

9663 are not necessarily equivalent; the latter is not specified by POSIX. 2 and may 2
9664 result in an error. The construct `\0` is used to mean ‘‘no separator’’ because his- 2
9665 torical versions of `paste` did not follow the syntax guidelines and the command 2

```
9666      paste -d"" . . .                                       2
```

9667 could not be handled properly by *getopt* (). 2

9668 **History of Decisions Made**

9669 Because most of the standards utilities work on text files, `cut` and `paste` are
9670 required to process lines of arbitrary length as a means of converting long lines
9671 from arbitrary sources into text files and converting processed text files back into
9672 files with arbitrary line lengths to interface with those applications that require
9673 long lines as input.

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

9674 ## 4.47 pathchk — Check pathnames

9675 ### 4.47.1 Synopsis

9676 `pathchk [-p]` *pathname* ...

9677 ### 4.47.2 Description

9678 The `pathchk` utility shall check that one or more pathnames are valid (i.e., they
9679 could be used to access or create a file without causing syntax errors) and portable
9680 (i.e., no filename truncation will result). More extensive portability checks are
9681 provided by the `-p` option.

9682 By default, the `pathchk` utility shall check each component of each *pathname*
9683 operand based on the underlying file system. A diagnostic shall be written for
9684 each *pathname* operand that:

9685 — is longer than {PATH_MAX} bytes (see Pathname Variable Values in
9686 POSIX.1 {8}2.9.5),

9687 — contains any component longer than {NAME_MAX} bytes in its containing
9688 directory,

9689 — contains any component in a directory that is not searchable, or

9690 — contains any character in any component that is not valid in its containing
9691 directory.

9692 The format of the diagnostic message is not specified, but shall indicate the error
9693 detected and the corresponding *pathname* operand.

9694 It shall not be considered an error if one or more components of a *pathname*
9695 operand do not exist as long as a file matching the pathname specified by the
9696 missing components could be created that does not violate any of the checks
9697 specified above.

9698 ### 4.47.3 Options

9699 The `pathchk` utility shall conform to the utility argument syntax guidelines
9700 described in 2.10.2.

9701 The following option shall be supported by the implementation:

9702 `-p` Instead of performing checks based on the underlying file system,
9703 write a diagnostic for each *pathname* operand that:

9704 — is longer than {_POSIX_PATH_MAX} bytes (see Minimum
9705 Values in POSIX.1 {8}2.9.2),

9706 — contains any component longer than {_POSIX_NAME_MAX}
9707 bytes, or

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

— contains any character in any component that is not in the portable filename character set (see 2.2.2.111).

4.47.4 Operands

The following operand shall be supported by the implementation:

pathname A pathname to be checked.

4.47.5 External Influences

4.47.5.1 Standard Input

None.

4.47.5.2 Input Files

None.

4.47.5.3 Environment Variables

The following environment variables shall affect the execution of `pathchk`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

LC_MESSAGES This variable shall determine the language in which messages should be written.

4.47.5.4 Asynchronous Events

Default.

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

4.47.6 External Effects

4.47.6.1 Standard Output

None.

4.47.6.2 Standard Error

Used only for diagnostic messages.

4.47.6.3 Output Files

None.

4.47.7 Extended Description

None.

4.47.8 Exit Status

The `pathchk` utility shall exit with one of the following values:

0 All *pathname* operands passed all of the checks.

>0 An error occurred.

4.47.9 Consequences of Errors

Default.

4.47.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

To verify that all pathnames in an imported data interchange archive are legitimate and unambiguous on the current system:

```
pax -f archive | xargs pathchk
if [ $? -eq 0 ]
then
        pax -r -f archive
else
        echo Investigate problems before importing files.
        exit 1
fi
```

To verify that all files in the current directory hierarchy could be moved to any POSIX .1 {8}conforming system that also supports the `pax` utility:

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

```
find . -print | xargs pathchk -p
if [ $? -eq 0 ]
then
        pax -w -f archive .
else
        echo Portable archive cannot be created.
        exit 1
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:

```
case $- in
        *C*)    reset="";;
        *)      reset="set +C"
                set -C;;
esac
test -r "$path" && pathchk "$path.out" &&
        rm "$path.out" > "$path.out"
if [ $? -ne 0 ]; then
        printf "%s: %s not found or %s.out fails \
creation checks.\n" $0 "$path" "$path"
        $reset  # reset the noclobber option in case a trap
                # on EXIT depends on it
        exit 1
fi
$reset
PROCESSING < "$path" > "$path.out"
```

The following assumptions are made in this example:

(1) `PROCESSING` represents the code that will be used by the application to use `$path` once it is verified that `$path.out` will work as intended.

(2) The state of the *noclobber* option is unknown when this code is invoked and should be set on exit to the state it was in when this code was invoked. (The `reset` variable is used in this example to restore the initial state.)

(3) Note the usage of `rm "$path.out" > "$path.out"`:

(a) The `pathchk` command has already verified, at this point, that `$path.out` will not be truncated.

(b) With the *noclobber* option set, the shell will verify that `$path.out` does not already exist before invoking `rm`.

(c) If the shell succeeded in creating `$path.out`, `rm` will remove it so that the application can create the file again in the `PROCESSING` step.

(d) If the `PROCESSING` step wants the file to already exist when it is invoked, the

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

```
rm "$path.out" > "$path.out"
```

should be replaced with

```
> "$path.out"
```

which will verify that the file did not already exist, but leave `$path.out` in place for use by PROCESSING.

History of Decisions Made

The `pathchk` utility is new, commissioned for this standard. It, along with the `set –C` (*noclobber*) option added to the shell, replaces the `mktemp`, `validfnam`, and `create` utilities that appeared in earlier drafts. All of these utilities were attempts to solve a few common problems:

— Verify the validity (for several different definitions of ‘‘valid’’) of a pathname supplied by a user, generated by an application, or imported from an external source,

— Atomically create a file, and

— Perform various string handling functions to generate a temporary file name.

The `test` utility (see 4.62) can be used to determine if a given pathname names an existing file; it will not, however, give any indication of whether or not any component of the pathname was truncated in a directory where the {_POSIX _NO _TRUNC} feature (see Execution-Time Symbolic Constants for Portability Specification in POSIX.1 {8} 2.9.4) is not in effect. The `pathchk` utility provided here does not check for file existence; it performs checks to determine if a pathname does exist or could be created with no pathname component truncation.

The *noclobber* option added to the shell (see 3.14.11) can be used to atomically create a file. As with all file creation semantics in POSIX.1 {8}, 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. The `create` utility, included in one earlier draft, provided checking and atomic creation in a single invocation of the utility; these are orthogonal issues and need not be grouped into a single utility. Note that the *noclobber* option also provides a way of creating a lock for process synchronization; since it provides an atomic `create`, there is no race between a test for existence and the following creation if it did not exist.

Having a function like *tmpnam* () in the C Standard {7} is important in many high-level languages. The shell programming language, however, has built-in string manipulation facilities, making it very easy to construct temporary file names. The names needed obviously depend on the application, but are frequently of a form similar to

`$TMPDIR/`*application_abbreviation* `$$.`*suffix*

In cases where there is likely to be contention for a given suffix, a simple shell `for` or `while` loop can be used with the shell *noclobber* option to create a file without risk of collisions, as long as applications trying to use the same filename

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

namespace are cooperating on the use of files after they have been created.

4.48 pax — Portable archive interchange

4.48.1 Synopsis

```
pax [–cdnv][–f archive ][–s replstr ]... [pattern ...]
pax –r[–cdiknuv][–f archive ][–o options ]... [–p string ]... [–s replstr ]
    ... [pattern ...]
pax –w[–dituvX][–b blocksize ][ [–a][–f archive ]][–o options ]...
    [–s replstr ]... [–x format ][file ...]
pax –r –w[–diklntuvX][–p string ]... [–s replstr ]... [file ...] directory
```

4.48.2 Description

The pax utility shall read, write, and write lists of the members of archive files and copy directory hierarchies. A variety of archive formats shall be supported; see the –x *format* option description under 4.48.3.

The action to be taken depends on the presence of the –r and –w options:

(1) When neither the –r option nor the –w option is specified, pax shall write the names of the members of the archive file read from the standard input, with pathnames matching the specified patterns, to standard output. If a named file is of type directory, the file hierarchy rooted at that file shall be written out as well.

(2) When the –r option is specified, but the –w option is not, pax shall extract the members of the archive file read from the standard input, with pathnames matching the specified patterns. If an extracted file is of type directory, the file hierarchy rooted at that file shall be extracted as well. The extracted files shall be created relative to the current file hierarchy.

The ownership, access and modification times, and file mode of the restored files are discussed under the –p option.

(3) When the –w option is specified and the –r option is not, pax shall write the contents of the file operands to the standard output in an archive format. If no *file* operands are specified, a list of files to copy, one per line, shall be read from the standard input. A file of type directory shall include all of the files in the file hierarchy rooted at the file.

(4) When both the –r and –w options are specified, pax shall copy the file operands to the destination directory.

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

If no *file* operands are specified, a list of files to copy, one per line, shall be read from the standard input. A file of type directory shall include all of the files in the file hierarchy rooted at the file.

The effect of the copy shall be as if the copied files were written to an archive file and then subsequently extracted, except that there may be hard links between the original and the copied files. If the destination directory is a subdirectory of one of the files to be copied, the results are unspecified. If the destination directory is a file of a type not defined by POSIX.1 {8}, the results are implementation defined; otherwise it shall be an error for the file named by the directory operand not to exist, not be writable by the user, or not be a file of type directory.

If, when the `-r` option is specified, intermediate directories are necessary to extract an archive member, `pax` shall perform actions equivalent to the POSIX.1 {8} *mkdir* () function, called with the following arguments:

— The intermediate directory used as the *path* argument.

— The value of the bitwise inclusive OR of S_IRWXU, S_IRWXG, and S_IRWXO as the *mode* argument.

If any specified *pattern* or *file* operands are not matched by at least one file or archive member, `pax` shall write a diagnostic message to standard error for each one that did not match and exit with a nonzero exit status.

The supported archive formats shall be automatically detected on input. The default output archive format shall be implementation defined.

A single archive can span multiple files. The `pax` utility shall determine, in an implementation-defined manner, what file to read or write as the next file.

If the selected archive format supports the specification of linked files, it shall be an error if these files cannot be linked when the archive is extracted. Any of the various names in the archive that represent a file can be used to select the file for extraction.

4.48.3 Options

The `pax` utility shall conform to the utility argument syntax guidelines described in 2.10.2, except that the order of presentation of the `-s` options is significant.

The following options shall be supported by the implementation:

`-r` Read an archive file from standard input.

`-w` Write files to the standard output in the specified archive format.

`-a` Append files to the end of the archive. It is implementation defined which devices on the system support appending. Additional file formats unspecified by this standard may impose restrictions on appending.

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

9921 `-b` *blocksize*
9922 Block the output at a positive decimal integer number of bytes per
9923 write to the archive file. Devices and archive formats may impose
9924 restrictions on blocking. Blocking shall be automatically deter-
9925 mined on input. Conforming POSIX.2 applications shall not
9926 specify a *blocksize* value larger than 32 256. Default blocking
9927 when creating archives depends on the archive format. (See the
9928 `-x` option below.)

9929 `-c` Match all file or archive members except those specified by the
9930 *pattern* or *file* operands.

9931 `-d` Cause files of type directory being copied or archived or archive
9932 members of type directory being extracted to match only the file
9933 or archive member itself and not the file hierarchy rooted at the
9934 file.

9935 `-f` *archive* Specify the pathname of the input or output archive, overriding
9936 the default standard input (when neither the `-r` option nor the `-w`
9937 option is specified, or the `-r` option is specified and the `-w` option
9938 is not) or standard output (when the `-w` option is specified and the
9939 `-r` option is not).

9940 `-i` Interactively rename files or archive members. For each archive
9941 member matching a *pattern* operand or file matching a *file*
9942 operand, a prompt shall be written to the file `/dev/tty`. The
9943 prompt shall contain the name of the file or archive member, but
9944 the format is otherwise unspecified. A line shall then be read
9945 from `/dev/tty`. If this line is blank, the file or archive member
9946 shall be skipped. If this line consists of a single period, the file or
9947 archive member shall be processed with no modification to its
9948 name. Otherwise, its name shall be replaced with the contents of
9949 the line. The `pax` utility shall immediately exit with a nonzero
9950 exit status if end-of-file is encountered when reading a response
9951 or if `/dev/tty` cannot be opened for reading and writing.

9952 `-k` Prevent the overwriting of existing files.

9953 `-l` (The letter ell.) Link files. When both the `-r` and `-w` options are
9954 specified, hard links shall be made between the source and desti-
9955 nation file hierarchies whenever possible.

9956 `-n` Select the first archive member that matches each *pattern*
9957 operand. No more than one archive member shall be matched for
9958 each pattern (although members of type directory shall still
9959 match the file hierarchy rooted at that file).

9960 `-o` *options* Provide information to the implementation to modify the algo-
9961 rithm for extracting or writing files that is specific to the file for-
9962 mat specified by `-x`. This version of this standard does not
9963 specify any such options and a Strictly Conforming POSIX.2 Appli-
9964 cation shall not use the `-o` option.

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

NOTE: It is expected that future versions of POSIX.2 will offer additional file formats and this option will be used by POSIX.2 and other POSIX standards to specify such features as international file-name and file codeset translations, security, accounting, etc., related to each additional format.

`-p` *string* Specify one or more file characteristic options (privileges). The *string* option-argument shall be a string specifying file characteristics to be retained or discarded on extraction. The string shall consist of the specification characters `a`, `e`, `m`, `o`, and `p`, and/or other, implementation-defined, characters. Multiple characteristics can be concatenated within the same string and multiple `-p` options can be specified. The meaning of the specification characters are as follows:

- `a` Do not preserve file access times.
- `e` Preserve the user ID, group ID, file mode bits (see 2.2.2.60), access time, modification time, and any other, implementation-defined, file characteristics.
- `m` Do not preserve file modification times.
- `o` Preserve the user ID and group ID.
- `p` Preserve the file mode bits. Other, implementation-defined file-mode attributes may be preserved.

In the preceding list, ''preserve'' indicates that an attribute stored in the archive shall be given to the extracted file, subject to the permissions of the invoking process; otherwise, the attribute shall be determined as part of the normal file creation action (see 2.9.1.4).

If neither the `e` nor the `o` specification character is specified, or the user ID and group ID are not preserved for any reason, `pax` shall not set the S_ISUID and S_ISGID bits of the file mode.

If the preservation of any of these items fails for any reason, `pax` shall write a diagnostic message to standard error. Failure to preserve these items shall affect the final exit status, but shall not cause the extracted file to be deleted.

If file-characteristic letters in any of the *string* option-arguments are duplicated or conflict with each other, the one(s) given last shall take precedence. For example, if `-p eme` is specified, file modification times shall be preserved.

`-s` *replstr* Modify file or archive member names named by *pattern* or *file* operands according to the substitution expression *replstr*, using the syntax of the `ed` utility (see 4.20). The concepts of ''address'' and ''line'' are meaningless in the context of the `pax` utility, and shall not be supplied. The format shall be:

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

10006 −s /*old*/*new*/[gp]

10007 where as in ed, *old* is a basic regular expression and *new* can con-
10008 tain an ampersand, \ *n* (where *n* is a digit) backreferences, or
10009 subexpression matching. The *old* string shall also be permitted to
10010 contain <newline> characters.

10011 Any nonnull character can be used as a delimiter (/ shown here).
10012 Multiple −s expressions can be specified; the expressions shall be
10013 applied in the order specified, terminating with the first success-
10014 ful substitution. The optional trailing g shall be as defined in the
10015 ed utility. The optional trailing p shall cause successful substitu-
10016 tions to be written to standard error. File or archive member
10017 names that substitute to the empty string shall be ignored when
10018 reading and writing archives.

10019 −t Cause the access times of the archived files to be the same as they
10020 were before being read by pax.

10021 −u Ignore files that are older (having a less recent file modification
10022 time) than a pre-existing file or archive member with the same
10023 name. If the −r option is specified and the −w option is not
10024 specified, an archive member with the same name as a file in the
10025 file system shall be extracted if the archive member is newer than
10026 the file. If the −w option is specified and the −r option is not
10027 specified, an archive file member with the same name as a file in
10028 the file system shall be superseded if the file is newer than the
10029 archive member. It is unspecified if this is accomplished by
10030 actual replacement in the archive or by appending to the archive.
10031 If both the −r and −w options are specified, the file in the destina-
10032 tion hierarchy shall be replaced by the file in the source hierarchy
10033 or by a link to the file in the source hierarchy if the file in the
10034 source hierarchy is newer.

10035 −v Produce a verbose table of contents (see 4.48.6.1) if neither the −r
10036 option nor the −w option is specified. Otherwise, list archive
10037 member pathnames to standard error (see 4.48.6.2).

10038 −x *format* Specify the output archive format. The pax utility shall recognize
10039 the following formats:

10040 cpio The extended cpio interchange format specified in
10041 POSIX .1 {8} 10.1.2. The default *blocksize* for this for- 1
10042 mat for character special archive files shall be 5120. 1
10043 Implementations shall support all *blocksize* values 1
10044 less than or equal to 32 256 that are multiples of 1
10045 512.

10046 ustar The extended tar interchange format specified in
10047 POSIX .1 {8} 10.1.1. The default *blocksize* for this for- 1
10048 mat for character special archive files shall be 1
10049 10 240. Implementations shall support all *blocksize* 1

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

values less than or equal to 32 256 that are multiples of 512.

Implementation-defined formats shall specify a default block size as well as any other block sizes supported for character special archive files.

Any attempt to append to an archive file in a format different from the existing archive format shall cause `pax` to exit immediately with a nonzero exit status.

`-X` When traversing the file hierarchy specified by a pathname, `pax` shall not descend into directories that have a different device ID [*st_dev* , see POSIX .1 {8}*stat* ()].

The options that operate on the names of files or archive members (`-c`, `-i`, `-n`, `-s`, `-u`, and `-v`) shall interact as follows. When the `-r` option is specified and the `-w` option is not (archive members are being extracted), the archive members shall be ''selected,'' based on the user-specified *pattern* operands as modified by the `-c`, `-n`, and `-u` options. Then, any `-s` and `-i` options shall modify, in that order, the names of the selected files. The `-v` option shall write names resulting from these modifications.

When the `-w` option is specified (files are being archived), the files shall be selected based on the user-specified pathnames as modified by the `-n` and `-u` options. Then, any `-s` and `-i` options shall, in that order, modify the names of these selected files. The `-v` option shall write names resulting from these modifications.

If both the `-u` and `-n` options are specified, `pax` shall not consider a file selected unless it is newer than the file to which it is compared.

4.48.4 Operands

The following operands shall be supported by the implementation:

directory The destination directory pathname for copies when both the `-r` and `-w` options are specified.

file A pathname of a file to be copied or archived.

pattern A pattern matching one or more pathnames of archive members. A pattern shall be given in the name-generating notation of the pattern matching notation in 3.13, including the filename expansion rules in 3.13.3. The default, if no *pattern* is specified, is to select all members in the archive.

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

10085 ### 4.48.5 External Influences

10086 #### 4.48.5.1 Standard Input

10087 If the `-w` option is specified, the standard input shall be used only if no *file*
10088 operands are specified. It shall be a text file containing a list of pathnames, one
10089 per line, without leading or trailing <blank>s.

10090 If neither the `-f` nor `-w` options are specified, the standard input shall be an
10091 archive file. (See 4.48.5.2.)

10092 Otherwise, the standard input shall not be used.

10093 #### 4.48.5.2 Input Files

10094 The input file named by the *archive* option-argument, or standard input when the
10095 archive is read from there, shall be a file formatted according to one of the
10096 specifications in POSIX.1 {8}10.1, or some other, implementation-defined, format.

10097 The file `/dev/tty` shall be used to write prompts and read responses.

10098 #### 4.48.5.3 Environment Variables

10099 The following environment variables shall affect the execution of `pax`:

10100 **LANG** This variable shall determine the locale to use for the
10101 locale categories when both **LC_ALL** and the correspond-
10102 ing environment variable (beginning with **LC_**) do not
10103 specify a locale. See 2.6.

10104 **LC_ALL** This variable shall determine the locale to be used to over-
10105 ride any values for locale categories specified by the set-
10106 tings of **LANG** or any environment variables beginning
10107 with **LC_**.

10108 **LC_COLLATE** This variable shall determine the locale for the behavior of
10109 ranges, equivalence classes, and multicharacter collating
10110 elements used in the pattern matching expressions for the
10111 *pattern* operand, the basic regular expression for the `-s`
10112 option, and the extended regular expression defined for
10113 the `yesexpr` locale keyword in the LC_MESSAGES
10114 category.

10115 **LC_CTYPE** This variable shall determine the locale for the interpreta-
10116 tion of sequences of bytes of text data as characters (e.g.,
10117 single- versus multibyte characters in arguments and
10118 input files) and the behavior of character classes within
10119 regular expressions and pattern matching.

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

LC_MESSAGES This variable shall determine the processing of affirmative responses and the language in which messages should be written.

LC_TIME This variable shall determine the format and contents of date and time strings when the `-v` option is specified.

4.48.5.4 Asynchronous Events

Default.

4.48.6 External Effects

4.48.6.1 Standard Output

If the `-w` option is specified and neither the `-f` nor `-r` options are specified, the standard output shall be the archive formatted according to one of the specifications in POSIX.1 {8} 10.1, or some other implementation-defined format. (See `-x` *format* under 4.48.3.)

If neither the `-r` option nor the `-w` option is specified, the table of contents of the selected archive members shall be written to standard output using the following format:

`"%s\n"`, *<pathname>*

If neither the `-r` option nor the `-w` option is specified, but the `-v` option is specified, the table of contents of the selected archive members shall be written to standard output using the following formats:

For pathnames representing hard links to previous members of the archive:

`"%s␣==␣%s\n"`, *<ls -l listing>* , *<linkname>*

For all other pathnames:

`"%s\n"`, *<ls -l listing>*

where *<ls -l listing>* shall be the format specified by the `ls` utility (see 4.39) with the `-l` 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.

When writing a table of contents of selected archive members, standard output shall not be buffered more than a line at a time.

4.48.6.2 Standard Error

If either or both of the `-r` option and the `-w` option are specified as well as the `-v` option, `pax` shall write the pathnames it processes to the standard error output using the following format:

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

10155 "%s\n", <*pathname*>

10156 These pathnames shall be written as soon as processing is begun on the file or
10157 archive member, and shall be flushed to standard error. The trailing <newline>,
10158 which shall not be buffered, shall be written when the file has been read or
10159 written.

10160 If the −s option is specified, and the replacement string has a trailing p, substitu-
10161 tions shall be written to standard error in the following format:

10162 "%sΔ>>Δ%s\n", <*original pathname*> , <*new pathname*> 2

10163 In all operating modes of pax (see 4.48.2), optional messages of unspecified format
10164 concerning the input archive format and volume number, the number of files,
10165 blocks, volumes, and media parts as well as other diagnostic messages may be
10166 written to standard error.

10167 In all formats, for both standard output and standard error, it is unspecified how
10168 nonprintable characters in pathnames or linknames are written.

10169 **4.48.6.3 Output Files**

10170 If the −r option is specified, the extracted or copied output files shall be of the
10171 archived file type.

10172 If the −w option is specified, but the −r option is not, the output file named by the
10173 −f option argument shall be a file formatted according to one of the specifications
10174 in POSIX.1 {8}10.1, or some other, implementation-defined, format.

10175 **4.48.7 Extended Description**

10176 None.

10177 **4.48.8 Exit Status**

10178 The pax utility shall exit with one of the following values:

10179 0 All files were processed successfully.

10180 >0 An error occurred.

10181 **4.48.9 Consequences of Errors**

10182 If pax cannot create a file or a link when reading an archive or cannot find a file
10183 when writing an archive, or cannot preserve the user ID, group ID, or file mode
10184 when the −p option is specified, a diagnostic message shall be written to standard
10185 error and a nonzero exit status shall be returned, but processing shall continue.
10186 In the case where pax cannot create a link to a file, pax shall not, by default,
10187 create a second copy of the file.

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

If the extraction of a file from an archive is prematurely terminated by a signal or error, `pax` may have only partially extracted the file or (if the `-n` option was not specified) may have extracted a file of the same name as that specified by the user, but which is not the file the user wanted. Additionally, the file modes of extracted directories may have additional bits from the S_IRWXU mask set as well as incorrect modification and access times.

4.48.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

The following command:

```
pax -w -f /dev/rmt/1m .
```

copies the contents of the current directory to tape drive 1, medium density (assuming historical System V device naming procedures. The historical BSD device name would be `/dev/rmt9`).

The following commands:

```
mkdir newdir
pax -rw olddir newdir
```

copy the *olddir* directory hierarchy to *newdir* .

```
pax -r -s ',^//*usr//*,,' -f a.pax
```

reads the archive `a.pax`, with all files rooted in "`/usr`" in the archive extracted relative to the current directory.

The `-p` (privileges) option was invented to reconcile differences between historical `tar` and `cpio` implementations. In particular, the two utilities used `-m` in diametrically opposed ways. The `-p` option also provides a consistent means of extending the ways in which future file attributes can be addressed, such as for enhanced security systems or high-performance files. Although it may seem complex, there are really two modes that will be most commonly used:

`-p e` "Preserve everything." This would be used by the historical superuser, someone with all the appropriate privileges, to preserve all aspects of the files as they are recorded in the archive. The `e` flag is the sum of `o` and `p`, and other implementation-defined attributes.

`-p p` "Preserve" the file mode bits. This would be used by the user with regular privileges who wished to preserve aspects of the file other than the ownership. The file times are preserved by default, but two other flags are offered to disable these and use the time of extraction.

History of Decisions Made

The description of `pax` was adopted from a command written by Glenn Fowler of AT&T. It is a new utility, commissioned for this standard.

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

The table of contents output is written to standard output to facilitate pipeline processing.

The output archive formats required are those defined in POSIX.1 {8};others, such as the historical `tar` format, may be added as an extension.

The one pathname per line format of standard input precludes pathnames containing <newline>s. Although such pathnames violate the portable filename guidelines, they may exist and their presence may inhibit usage of `pax` within shell scripts. This problem is inherited from historical archive programs. The problem can be avoided by listing filename arguments on the command line instead of on standard input.

An earlier draft had hard links displaying for all pathnames. This was removed because it complicates the output of the non `-v` case and does not match historical `cpio` usage. The hard-link information is available in the `-v` display.

The working group realizes that the presence of symbolic links will affect certain `pax` operations. Historical practice, in both System V and BSD-based systems, is that the physical traversal of the file hierarchy shall be the default, and an option is provided to cause the utility to do a logical traversal, that is, follow symbolic links. Historical practice has not been so consistent as to what option is used to cause the logical traversal; BSD systems have used `-h` (`cp` and `tar`) and `-L` (`ls`), while the *SVID* specifies `-L` (`cpio` and `ls`). Given this inconsistency, the `-L` option is recommended.

The archive formats described in POSIX.1 {8} have certain restrictions that have been brought along from historical usage. For example, there are restrictions on the length of pathnames stored in the archive. When `pax` is used in `-rw` mode, copying directory hierarchies, there is no stated dependency on these archive formats. Therefore, such restrictions should not apply.

The POSIX.2 working group is currently devising a new archive format to be published in a revision or amendment to this standard. It is expected that the `ustar` and `cpio` formats then will be retired from a future version of POSIX.1 {8}. This new format will address all restrictions and new requirements for security labeling, etc. The `pax` utility should be upward-compatible enough to handle any such changes. The reason that the default `-x` *format* output format is implementation defined is to reserve the default format for this new standard interface. The `-o` option was devised to provide means of controlling the many aspects of international and security concerns without expending the entire alphabet of option letters for this, and possibly other, file formats. The `-o` string is meant to be specific for each `-x` format. Control of various file permissions and attributes that can be expressed in a binary way will continue to use the `-p` (permissions) option; the `-o` will be reserved for more involved requirements and will probably take a

```
pax -o name=value,name=value -o name=value
```

approach.

The fundamental difference in how `cpio` and `tar` viewed the world was in the way directories were treated. The `cpio` utility did not treat directories differently from other files, and to select a directory and its contents required that each file

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

in the hierarchy be explicitly specified. For `tar`, a directory matched every file in the file hierarchy it rooted.

The `pax` utility offers both interfaces; by default, directories map into the file hierarchy they root. The `-d` option causes `pax` to skip any file not explicitly referenced, as `cpio` traditionally did. The `tar`-style behavior was chosen as the default because it was believed that this was the more common usage, and because `tar` is the more commonly available interface, as it was historically provided on both System V and BSD implementations. Because a file may be matched more than once without causing it to be selected multiple times, the traditional usage of piping an `ls` or `find` to the archive command works as always.

The Data Interchange Format specification of POSIX.1 {8} requires that processes with ''appropriate privileges'' shall always restore the ownership and permissions of extracted files exactly as archived. If viewed from the historic equivalence between super-user and ''appropriate privileges,'' there are two problems with this requirement. First, users running as super-users may unknowingly set dangerous permissions on extracted files. Second, it is needlessly limiting in that super-users cannot extract files and own them as super-user unless the archive was created by the super-user. (It should be noted that restoration of ownerships and permissions for the super-user, by default, is historical practice in `cpio`, but not in `tar`.) In order to avoid these two problems, the `pax` specification has an additional ''privilege'' mechanism, the `-p` option. Only a `pax` invocation with the POSIX.1 {8} privileges needed, and which has the `-p` option set using the `e` specification character, has the ''appropriate privilege'' to restore full ownership and permission information.

Note also that POSIX.1 {8} 10.1 requires that the file ownership and access permissions shall be set, on extraction, in the same fashion as the POSIX.1 {8} *creat* () function when provided the mode stored in the archive. This means that the file creation mask of the user is applied to the file permissions.

The default *blocksize* value of 5120 for `cpio` was selected because it is one of the standard block-size values for `cpio`, set when the `-B` option is specified. (The other default block-size value for `cpio` is 512, and this was felt to be too small.) The default block value of 10 240 for `tar` was selected as that is the standard block-size value for BSD `tar`. The maximum block size of 32 256 (2^{15}–512) is the largest multiple of 512 that fits into a signed 16-bit tape controller transfer register. There are known limitations in some historic system that would prevent larger blocks from being accepted. Historic values were chosen to make compatibility with existing scripts using `dd` or similar utilities to manipulate archives more likely. Also, default block sizes for any file type other than character special has been deleted from the standard as unimportant and not likely to affect the structure of the resulting archive.

Implementations are permitted to modify the block-size value based on the archive format or the device to which the archive is being written. This is to provide implementations the opportunity to take advantage of special types of devices, and should not be used without a great deal of consideration as it will almost certainly decrease archive portability.

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

The −n option in early drafts had three effects; the first was to cause special characters in patterns to not be treated specially. The second was to cause only the first file that matched a pattern to be extracted. The third was to cause pax to write a diagnostic message to standard error when no file was found matching a specified pattern. Only the second behavior is retained by POSIX. 2, for many reasons. First, it is in general a bad idea for a single option to have multiple effects. Second, the ability to make pattern matching characters act as normal characters is useful for other parts of pax than just file extraction. Third, a finer degree of control over the special characters is useful, because users may wish to normalize only a single special character in a single file name. Fourth, given a more general escape mechanism, the previous behavior of the −n option can be easily obtained using the −s option or a sed script. Finally, writing a diagnostic message when a pattern specified by the user is unmatched by any file is useful behavior in all cases.

There are two methods of copying subtrees in POSIX. 2. The other method is described as part of the cp utility (see 4.13). Both methods are historical practice: cp provides a simpler, more intuitive interface, while pax offers a finer granularity of control. Each provides additional functionality to the other; in particular, pax maintains the hard-link structure of the hierarchy, while cp does not. It is the intention of the working group that the results be similar (using appropriate option combinations in both utilities). The results are not required to be identical; there seemed insufficient gain to applications to balance the difficulty of implementations having to guarantee that the results would be exactly identical.

A single archive may span more than one file. See POSIX .1 {8} 10.1.3. While POSIX .1 {8} only refers to reading the archive file, it is reasonable that the format utility may also determine, in an implementation-defined manner, the next file to write. It is suggested that implementations provide informative messages to the user on the standard error whenever the archive file is changed.

The −d option (do not create intermediate directories not listed in the archive) found in previous drafts of this standard was originally provided as a complement to the historic −d option of cpio. It has been deleted.

The −s option in earlier drafts specified a subset of the substitution command from the ed utility. As there was no reason for only a subset to be supported, the −s option is now compatible with the current ed specification. Since the delimiter can be any nonnull character, the following usage with single spaces is valid:

```
pax -s " foo bar " ...
```

The −t option (specify an implementation-defined identifier naming an input or output device) found in earlier drafts has been deleted because it is not historical practice and of limited utility. In particular, historic versions of neither cpio nor tar had the concept of devices that were not mapped into the file system; if the devices are mapped into the file system, the −f option is sufficient.

The −o and −p options found in previous versions of this standard have been renamed to be −p and −t, respectively, to correspond more closely with the historic tar and cp utilities.

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