# IEEE P1003.2a Draft 8 – December 1991

collision with the `−o` from System V (which has been provided here as `-t o`).

The default output format of `nm` is not specified because of differences in historical implementations. The `P` option was added to satisfy balloting objections relating to the use of `nm` in scripts as well as interactively. Some type of portable format was required. The format given in 5.21.6.1 uses spaces between the fields, which may be any number of `<blank>`s required to get the columns to line up. The single-character types were selected to match historical practice and the requirement that implementation additions also be single characters make parsing the information easier for shell scripts.

## 5.22 `patch` — Apply changes to files

### 5.22.1 Synopsis

`patch` [`−blNR`] [ `−c` | `−e` | `−n` ] [`−d` *dir* ] [`−D` *define* ] [`−i` *patchfile* ] [`−o` *outfile* ] 8
[`−r` *rejectfile* ] [*file* ] 8

### 5.22.2 Description

The `patch` utility shall read a source (''patch'') file containing any of the three forms of difference (''diff'') listings produced by the `diff` (see 4.17) utility (normal, context, or in the style of `ed`; see 4.20) and apply those differences to a file. By default, `patch` shall read from the standard input.

The `patch` utility shall attempt to determine the type of the diff listing, unless overruled by a `−c`, `−e`, or `−n` option.

If the patch file contains more than one patch, `patch` shall attempt to apply each
of them as if they came from separate patch files. (In this case the name of the 7
patch file must be determinable for each `diff` listing.) 7

### 5.22.3 Options

The `patch` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

`−b` Save a copy of the original contents of each modified file, before 7
the differences are applied, in a file of the same name with the 7
suffix `.orig` appended to it. If the file already exists, it shall be 7
overwritten; if multiple patches are applied to the same file, the 7
`.orig` file shall be written only for the first patch. When the 8
`−o` *outfile* option is also specified, *file*`.orig` shall not be created 7
but, if *outfile* already exists, *outfile*`.orig` shall be created. 7

`−c` Interpret the patch file as a context diff (the output of the utility `diff` when the `−c` or `−C` options are specified).

`−d` *dir* Change the current directory to *dir* before processing as described in 5.22.7.

`−D` *define* Mark changes with the C preprocessor construct:

```
#ifdef define
...
#endif
```

The option-argument *define* shall be used as the differentiating symbol.

`−e` Interpret the patch file as an `ed` script, rather than a `diff` script.

`−i` *patchfile* Read the patch information from the file named by the pathname *patchfile*, rather than the standard input.

`−l` (The letter ell.) Cause any sequence of `<blank>`s in the diff script to match any sequence of `<blank>`s in the input file. Other characters shall be matched exactly.

`−n` Interpret the script as a normal diff.

`−N` Ignore patches where the differences have already been applied to the file; by default, already-applied patches shall be rejected.

`−o` *outfile* Instead of modifying the files (specified by the *file* operand or the diff listings) directly, write a copy of the file referenced by each patch, with the appropriate differences applied, to *outfile*. 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 *outfile*.

`−R` Reverse the sense of the patch script; i.e., assume that the diff script was created from the new version to the old version. The `−R` option cannot be used with `ed` scripts. The `patch` utility attempts 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, `patch` 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 if the `−R` option should be set.

`−r` *rejectfile* Override the default reject file name. In the default case, the reject file shall have the same name as the output file, with the suffix `.rej` appended to it. See 5.22.7.3.

### 5.22.4 Operands

The following operand shall be supported by the implementation:

*file* A pathname of a file to patch.

### 5.22.5 External Influences

#### 5.22.5.1 Standard Input

See Input Files.

#### 5.22.5.2 Input Files

Input files shall be text files.

#### 5.22.5.3 Environment Variables

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

**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 interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files).

**LC_MESSAGES** This variable shall determine the processing of affirmative 7
responses and the language in which messages should be 7
written. 7

**LC_TIME** This variable shall determine the locale for recognizing the format of file time stamps written by the `diff` utility in a context-diff input file.

#### 5.22.5.4 Asynchronous Events

Default.

### 5.22.6 External Effects

#### 5.22.6.1 Standard Output

None.

#### 5.22.6.2 Standard Error

Used for diagnostic and informational messages. 7

#### 5.22.6.3 Output Files

The output of the `patch` utility, the save files (`.orig` suffixes) and the reject files (`.rej` suffixes) shall be text files.

### 5.22.7 Extended Description

A patchfile may contain patching instructions for more than one file; file names 8
shall be determined as specified in 5.22.7.2. When the `−b` option is specified, for each patched file, the original shall be saved in a file of the same name with the suffix `.orig` appended to it.

For each patched file, a reject file may also be created as noted in 5.22.7.3. In the absence of a `−r` option, the name of this file shall be formed by appending the suffix `.rej` to the original file name.

#### 5.22.7.1 Patchfile Format

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 `diff −c` (see 4.17), and one or more sets of `diff` output, which are customarily called ''hunks.''

The `patch` utility shall recognize the following expression in the header information:

| | |
|---|---|
| `Index:` *pathname* | The file to be patched is named *pathname*. |

8

If all lines (including headers) within a patch begin with the same leading 7
sequence of `<blank>`s, the `patch` utility shall remove this sequence before 7
proceeding. Within each patch, if the type of diff is context, the `patch` utility shall recognize the following expressions:

| | |
|---|---|
| `***` *filename timestamp* | The patches arose from *filename.* |
| `---` *filename timestamp* | The patches should be applied to *filename.* |

Each hunk within a patch shall be the `diff` output to change a line range within the original file. The line numbers for successive hunks within a patch shall occur in ascending order.

#### 5.22.7.2 Filename Determination

If no *file* operand is specified, `patch` shall perform the following steps to obtain a pathname:

(1) If the patch contains the strings *** and `---`, the `patch` utility shall test 8
for the existence of each pathname relative to the current directory. If 8
both files exist, the `patch` utility shall assume that no pathname can be 7
obtained from this step. 7

(2) If the header information contains a line with the string `Index:`, the
`patch` utility shall test for the existence of this file relative to the current 8
directory. 8

(3) If no pathname can be obtained by applying the steps (1) or (2), or if the
pathnames obtained do not exist, the `patch` utility shall write a prompt 8
to standard output and request a filename interactively from standard 8
input.

#### 5.22.7.3 Patch Application

If the `−c`, `−e`, or `−n` option is present, the `patch` utility shall interpret information within each hunk as a context diff, an `ed` diff, or a normal diff, respectively. In the absence of any of these options, the `patch` utility shall determine the type of diff based on the format of information within the hunk.

8

For each hunk, the `patch` utility shall begin to search for the place to apply the
patch at the line number at the beginning of the hunk, plus or minus any offset
used in applying the previous hunk. If lines matching the hunk context are not 7
found, the `patch` utility shall scan both forwards and backwards at least 1000 7
bytes for a set of lines that match the hunk context. 7

If no such place is found, and it is a context diff, then another scan shall take place, ignoring the first and last line of context. If that fails, the first two and last two lines of context shall be ignored and another scan shall be made. Implementations may search more extensively for installation locations.

If no location can be found, the `patch` utility shall append the hunk to the reject 7
file. The rejected hunk shall be written in context-diff format regardless of the
format of the patch file. If the input was a normal or `ed`-style diff, the reject file 7
may contain diffs with zero lines of context. The line numbers on the hunks in the reject file may be different than the line numbers in the patch file since they shall reflect the approximate locations for the failed hunks in the new file rather than the old one.

If the type of patch is an `ed` diff, the implementation may accomplish the patching by invoking the `ed` utility.

### 5.22.8 Exit Status

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

- 0 Successful completion.
- 1 One or more lines were written to a reject file.
- >1 An error occurred.

### 5.22.9 Consequences of Errors

Patches that cannot be correctly placed in the file shall be written to a reject file.

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

#### Usage, Examples

The −R option will not work with `ed` scripts because there is too little information to reconstruct the reverse operation.

#### History of Decisions Made

Some of the functionality in historical `patch` implementations has not been specified. The following documents features present in historical implementations that have not been specified.

The `patch` utility's ability to ''check out'' or ''get'' suitable RCS or SCCS files if the source file is unavailable was deleted because POSIX.2 has not yet, if it ever will, required SCCS or RCS, and, more importantly, because it violates the ''Principle of Least Astonishment'' for users. It is strongly suggested that this functionality, if provided, be under the control of a command-line option.

Another deleted piece of functionality was the ''+'' pseudo-option allowing an additional set of options and patch file operand to be given. This was seen as being insufficiently useful to standardize.

In historical implementations, if the string `Prereq:` appeared in the header, the `patch` utility would search for corresponding version information (the string specified in the header, delimited by <blank>s or the beginning or end of a line or the file) anywhere in the original file. This was deleted as too simplistic and insufficiently trustworthy a mechanism to standardize. For example, if

```
Prereq: 1.2
```

were in the header, the presence of a delimited 1.2 anywhere in the file would satisfy the prerequisite.

The following options were dropped from historical implementations of `patch`, as insufficiently useful to standardize.

`−b` The `−b` option traditionally provided a method for changing the backup file's name extension from `.orig`. This option has been modified and retained in the current specification.

`−F` The `−F` option specified the number of lines of a context diff to ignore when searching for a place to install a patch.

`−f` The `−f` option traditionally caused `patch` to not request additional information from the user.

`−p` The `−p` option attempted to provide a method of customizing the patch file to local user file hierarchies.

Historical versions of `patch` resolved file names in a more complex fashion than that described by this standard. If no *file* operand was specified, the full pathnames and terminating filenames in any context diff as well as any pathnames specified using the `Index:` keyword in the patch header were checked for existence. If none of these files existed, the user was prompted for the name of a file to patch. Additionally, the `−p` option could be used to delete some of the pathname components from the context diff and `Index:` keyword pathnames. POSIX simplifies this behavior and does not require the `−p` option for two reasons. The first is that sites with strong naming and stylistic conventions, or with local and ''official'' copies of programs, could easily patch the wrong files with the user being unaware of the error. The second is that the `−p` option was only useful if the local tree was entirely consistent with the tree from which the patch file was created. This was believed unlikely, particularly in patch files with multiple patches, and the functionality is trivially replaced by creating the patch file at a more reasonable place in the tree.

`−r` The `−r` option traditionally provided a method of overriding the default reject file's extension from `.rej`.

`−s` The `−s` option traditionally caused `patch` to work silently unless an error occurred.

`−x` The `−x` option traditionally set internal debugging flags.

In some file system implementations, the saving of a `.orig` file may produce unwanted results. In the case of 12, 13, or 14-character filenames, on file systems supporting 14-character maximum filenames, the `.orig` file will overwrite the new file. The reject file may also exceed this filename limit. It was suggested, due to some existing practice, that a tilde (~) suffix be used instead of `.orig` and some other character instead of the `.rej` suffix. This was rejected because it is not obvious to the user which file is which. The suffixes `.orig` and `.rej` are clearer and more understandable.

The `−b` option has the opposite sense in some historical implementations—do not save the `.orig` file. The default case here is to not save the files, making `patch` behave more consistently with the other standard utilities.

The `−w` option in previous drafts was changed to `−l` to match historical practice.

The `−N` option was included because without it, a noninteractive application cannot reject previously applied patches. For example, if a user is piping the output of `diff` into the `patch` utility, and the user only wants to patch a file to a newer version noninteractively, the `−N` option is required.

Changes to the `−l` option description were proposed to allow matching across 8
`<newline>`s in addition to just `<blank>`s. Since this is not historical practice, 8
and since some ambiguities could result, it is suggested that future developments 8
in this area be triggered by another option letter, such as `−L`. 8

## 5.23 `ps` — Report process status

### 5.23.1 Synopsis

`ps [−aA] [−G` *grouplist* `] [−o` *format* `] . . . [−p` *proclist* `] [−t` *termlist* `]` 7
`[−U` *userlist* `]`

### 5.23.2 Description

The `ps` utility shall write information about processes, subject to having the appropriate privileges to obtain information about those processes.

When the `−o` option is not specified, information about processes selected shall be
written in an implementation-defined manner. By default, `ps` shall select all 7
processes with the same effective user ID as the current user and the same con- 7
trolling terminal as the invoker. In all cases, the selection of processes to be 7
reported shall be subject to the constraints of appropriate privileges. 7

### 5.23.3 Options

The `ps` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation.

`−a` Write information for all processes associated with terminals. 7
Implementations may omit session leaders from this list. 7

`−A` Write information for all processes.

`−G` *grouplist*
: Write information for processes whose real group ID numbers are given in *grouplist* . The *grouplist* shall be a single argument in the form of a `<blank>`- or comma-separated list.