

**NAME**

rename – renames multiple files

**SYNOPSIS**

```
rename [ -h|-m|-V ] [ -v ] [ -0 ] [ -n ] [ -f ] [ -d ] [ -e|-E perlexpr ]* [perlexpr [ files ]
```

**DESCRIPTION**

rename renames the filenames supplied according to the rule specified as the first argument. The *perlexpr* argument is a Perl expression which is expected to modify the `$_` string in Perl for at least some of the filenames specified. If a given filename is not modified by the expression, it will not be renamed. If no filenames are given on the command line, filenames will be read via standard input.

For example, to rename all files matching `*.bak` to strip the extension, you might say

```
rename 's/\.bak$//' *.bak
```

To translate uppercase names to lower, you'd use

```
rename 'y/A-Z/a-z/' *
```

**OPTIONS**

**-v, --verbose**

Verbose: print names of files successfully renamed.

**-0, --null**

Use `\0` as record separator when reading from STDIN.

**-n, --nono**

No action: print names of files to be renamed, but don't rename.

**-f, --force**

Over write: allow existing files to be over-written.

**--path, --fullpath**

Rename full path: including any directory component. DEFAULT

**-d, --filename, --nopath, --nofullpath**

Do not rename directory: only rename filename component of path.

**-h, --help**

Help: print SYNOPSIS and OPTIONS.

**-m, --man**

Manual: print manual page.

**-V, --version**

Version: show version number.

**-e**

Expression: code to act on files name.

May be repeated to build up code (like `perl -e`). If no `-e`, the first argument is used as code.

**-E**

Statement: code to act on files name, as `-e` but terminated by `';`.

**ENVIRONMENT**

No environment variables are used.

**AUTHOR**

Larry Wall

**SEE ALSO**

[mv\(1\)](#), [perl\(1\)](#)

**DIAGNOSTICS**

If you give an invalid Perl expression you'll get a syntax error.

**BUGS**

The original `rename` did not check for the existence of target filenames, so had to be used with care.