

NAME

changeall – Change all occurrences of one string to another in multiple files.

SYNOPSIS

`${HOME}/bin/changeall "frompattern" "topattern" file1 file2 ...`

DESCRIPTION

First the routine escapes the special characters `-$&/` if they are present in `frompattern` or `topattern`. Then it examines each file in the given list. If a file is not a nonempty regular file in this directory or if it is not permitted to write, it is skipped. If a file does not contain any occurrences of `frompattern` it is skipped. Otherwise the script uses `sed` to change all occurrences of `frompattern` to `topattern` in the file before moving on to the next one.

In the unlikely event that `sed` sets a nonzero return code in changing a target file, the script writes an error message, restores the original version of that file only and stops without processing the subsequent files. Original versions of the files that have been changed are left in `/tmp`.

UNITS and FILES

The listed files are changed, but only if they contain `frompattern`. This process involves copying the files into `/tmp`. The script writes informational and error messages on standard out.

DIAGNOSTICS

The script can stop with the following return codes.

- 0 `sed` reported no failure in modifying any target file
- 1 `sed` failed in modifying a target file

It is also possible for `sed` to fail in escaping a special character, in which case `sed` writes a message and interrupts the execution of this script.

BUGS

Escaping the special characters `-$&/` often allows the script to change strings containing them, but it might not be enough to prevent `sed` from being confused if you use them in combinations that are meaningful to `/bin/sh`. For example, `$*` will be expanded to the parameter list of the parent process rather than being treated as that character string. If you use characters that are not letters or numerals in either `frompattern` or `topattern`, you should test `changeall` on a single file to confirm that it does what you expect.

The program `${HOME}/bin/exe/change`, although less convenient to use, safely allows `frompattern` and `topattern` to contain characters that are meaningful to the shell.

EXAMPLE

```
unix[1] cat fyle
Here is a file containing -*$&/ some special characters.
unix[2] changeall "-*$&/ some" "no more" fyle
changing "-*$&/ some" to "no more"
changing fyle...done
unix[3] cat fyle
Here is a file containing no more special characters.
```

The example above changes the single line in fyle as shown. The example below changes "was" to "now" in every file in the current directory that contains "was". The example assumes that the string "was" is present in files named abc, def, ...

```
unix[4] changeall "was" "now" *
changing "was" to "now"
changing abc...done
changing def...done
:
```