rpl

NAME

rpl (RePLace) – replace strings in multiple files

VERSION

1.4.0

SYNOPSIS

rpl [-iwRspfdtx [-q|-v]] <old_str> <new_str> <target_file(s)>

DESCRIPTION

rpl replaces old_str with new_str in all target files. It returns the number of strings replaced or a system error code (non-zero) if there is an error.

Note that you should put strings in single quotes if they contain spaces. You must also escape all shell meta-characters. It's a good idea to put ALL strings in single quotes.

rpl will attempt to maintain the owner, group and permissions of your original files. For safety, *rpl* creates a temporary file and makes changes to that file. It then moves the temporary file over the original file. *rpl* sets the owner, group, and permissions of the new file to match those of the original file. In some circumstances *rpl* will not be able to do this (such as when a file is owned by the superuser but you have group write permission). In these cases *rpl* will warn you that the owner/group or permissions cannot be set and that file will be skipped, unless you use the force (-f) option. Note that the use of temp files in predictable, world-writeable locations could lead to symlink attacks. Ideally you should set the $TMPDIR environment variable to a private directory readable and writeable only by you. This is especially important if running *rpl* as root. You have been warned!

Normally, *rpl* will change the modification time of all files it processes like any other program. However, you may instruct *rpl* to keep the original modification times using the -d (Don't alter mod-times) option.

You can specify file suffixes to be searched using the -x option. Any files that do not match the specified suffixes will not be searched or modified. The -x option may be used more than once to tell *rpl* to search files with varying suffixes. For instance, say you wanted to search all of your ".html", ".htm", and ".php" files you would add

" -x'.html' -x'.htm' -x'.php' "

to your command line. *rpl* would then skip any files that did not end with these suffixes. This is mainly useful when doing recursive searching (-R option).

OPTIONS

–i Ignore case of old_str
rpl will match the old_str in the searched file regardless of the case. The case of new_str will not be altered.

–w Whole words (old_str bounded by white space in file)
rpl will only match old_str if it is bounded by the start of a line, a space, a tab, or the end of a line.

–q Quiet mode (no output at all)
Good for shell scripts, etc.

–v Verbose mode (lots of output)
rpl will list the name of each file and directory, and the line numbers that contain matches.

–R Search directories recursively
rpl will scan every file and every directory recursively. Without this option directories will be skipped.

–x Specify file suffixes to search. (e.g. ".html", ".c", etc.) May be used multiple times. See above for details.

–p Prompt for each file
rpl will prompt you before scanning each file. If you respond 'N' or 'n' *rpl* will skip that file and move on to the next file. The default action if you press enter is to process the file.

−s Simulation mode
rpl will scan all of the files and list the names of files that it would modify if a replace operation was executed. If you turn on the verbose (-v) option as well *rpl* will list the line numbers where the string is matched.

−e Honor Escapes
rpl will honor escape" sequences in old_string and new_string. Standard escapes such as "" (tab), "0 (newline), " (carriage return) are processed, as well as any octal or hexidecimal
ASCII codes. Octal ASCII codes start with a ’´ and are comprised of three digits [0-7] (e.g. ’ 15’). Hexidecimal ASCII codes start with ’ x’ followed by two characters [0-f] (e.g. ’ x0d’). The ’x’ and the [a-f] may be upper or lowercase. When you use this switch you must escape all backslash (’´) characters with another backslash (e.g. ’\’).

−f Force mode
rpl will overwrite files even if the owner, group, or permissions of the new file will not match the original. Obviously, *rpl* cannot overwrite files if the user does not have write permission.

−d Don’t change modification times
rpl will process files, but keep their original modification times.

−t Use $TMPDIR for temporary files
Causes *rpl* to write temporary files to the directory specified by the environment variable $TMPDIR instead of writing the temp files to the original file dir. (See note above about symlink attacks and temp files.)

−L Display the software license
This displays the software license that you agree to by using *rpl.*

−h Display a brief summary of options

BUGS

Report bugs to software@laffeycomputer.com

TODO

Grep pattern matching.

An option to backup the original file.

If you would like to see this or other enhancements send e-mail to software@laffeycomputer.com

CONTRIBUTORS

Original code:
Joe Laffey <software@laffeycomputer.com>

Other contributors include:
Devrim Erdem <devrim@infotron-tr.com>
Many thanks to the beta testers who sent in valuable feedback!