

Package: replace (via r-universe)

May 28, 2026

Title Replace Variable Names in R Scripts

Version 0.0.0.9000

Description Choose a target, a replacement, files or folders, and whether you want to replace standard variables, function calls, argument names, formal names, or package names.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports stringr, utils

URL <https://github.com/moodymudskipper/replace>

BugReports <https://github.com/moodymudskipper/replace/issues>

Config/pak/sysreqs libicu-dev

Repository <https://cynkra.r-universe.dev>

Date/Publication 2024-05-30 09:44:38 UTC

RemoteUrl <https://github.com/moodymudskipper/replace>

RemoteRef HEAD

RemoteSha 7deff4d7d968a4769d535c69aa59495c9f89e5dd

Contents

replace_in_files	2
Index	3

replace_in_files	<i>Replace occurrence of variable in files</i>
------------------	--

Description

Use to replace code in a collection of files, keeping control on the type of replacement.

Usage

```
replace_in_files(
  target,
  replacement,
  paths = "R",
  what = c("var", "fun"),
  regex = FALSE,
  ...
)
```

Arguments

target	a string
replacement	a string
paths	a vector of paths or folders, if the latter all R scripts (".R", ".r") are considered
what	one or several of "var", "fun", "arg", "formal", "package", "comment", "expression". They are mapped to token types found in the output of <code>getParseData()</code> . See details.
regex	a boolean. Whether to use regular expressions for detection and replacement. <ul style="list-style-type: none"> • "var": standard variable names <code>target</code>, includes element names <code>foo\$target</code> • "fun": function calls <code>target()</code> (if called through <code>lapply</code> etc a function will be matched with "var", not "fun") • "arg": Function argument name <code>foo(target = bar)</code> • "formal": a function's formal definition <code>function(target) {}</code> or <code>function(target =) {}</code> • "package": <code>target::foo</code> or <code>target:::foo</code> • "comment": comments, usually more useful with <code>regex = TRUE</code> • "string": strings like <code>"target"</code> • "expression": complex calls

For comments the "#" should be included in `target` and in the replacement and for strings, the quotes should be included in `target` (unless we use `regex = TRUE`).

Index

`replace_in_files`, 2