

# Package ‘tergo’

July 22, 2025

**Title** Style Your Code Fast

**Version** 0.1.9

**Description** Provides a set of functions that allow users for styling their R code according to the 'tidyverse' style guide. The package uses a native Rust implementation to ensure the highest performance. Learn more about 'tergo' at <<https://rtergo.pagacz.io>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Config/rextendr/version** 0.3.1.9001

**SystemRequirements** Cargo (Rust's package manager), rustc

**NeedsCompilation** yes

**Biarch** true

**Depends** R (>= 4.2)

**Suggests** rextendr (== 0.3.1), roxygen2, rstudioapi (>= 0.7), devtools, pkgdown, desc, knitr, rmarkdown, testthat (>= 3.0.0)

**URL** <https://rtergo.pagacz.io>, <https://github.com/kpagacz/tergo>

**BugReports** <https://github.com/kpagacz/tergo/issues>

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Author** Konrad Pagacz [aut, cre],  
Maciej Nasinski [ctb],  
The authors of the dependency Rust crates [ctb] (see inst/AUTHORS file for details)

**Maintainer** Konrad Pagacz <konrad.pagacz@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-04-11 16:40:01 UTC

Contents

|                              |          |
|------------------------------|----------|
| get_default_config . . . . . | 2        |
| style . . . . .              | 3        |
| style_file . . . . .         | 4        |
| style_pkg . . . . .          | 4        |
| style_text . . . . .         | 6        |
| <b>Index</b>                 | <b>7</b> |

---

|                    |                                      |
|--------------------|--------------------------------------|
| get_default_config | <i>Get the default configuration</i> |
|--------------------|--------------------------------------|

---

Description

This configuration is used by the styling functions if no value is provided for the configuration keys. It can also serve as the base for you custom configuration.

Usage

```
get_default_config()
```

Details

The configuration values:

- indent - the number of spaces to use for indentation.
- line\_length - the maximum number of characters in a line.
- embracing\_op\_no\_nl - whether to allow a newline after an embracing operator.
- allow\_nl\_after\_assignment - whether to allow a newline after an assignment operator.
- space\_before\_complex\_rhs\_in\_formula - whether to add a space before a complex right-hand side in a formula.
- strip\_suffix\_whitespace\_in\_function\_defs - whether to strip suffix whitespace in function definitions.
- function\_line\_breaks - the type of line breaks in function definitions when arguments do not fit. Possible values are: hanging, double, single.
- insert\_newline\_in\_quote\_call - whether to insert a newline in calls to quote.

Value

list with the default configuration

### Examples

```
config <- get_default_config()
print(config)

# Make the indent 4 spaces
config$indent <- 4L

# Make the maximum line length 80 characters
config$line_length <- 80L

# Make the function line breaks double
config$function_line_breaks <- "double"
```

---

style

*Style a package*

---

### Description

Style a package

### Usage

```
style(config_file = "tergo.toml", configuration = list(), ...)
```

### Arguments

`config_file` (character) The path to the configuration file. Default "tergo.toml".  
`configuration` (list) Configuration for formatting. Default `list()`.  
... additional parameters to `style_pkg()`

### Details

Configuration is read from a file named `tergo.toml` in the root of the package. The precedence of the configuration is (from the highest to lowest):

1. The configuration passed to the function.
2. The configuration file.

To see possible configuration options, see `get_default_config()`.

### Value

No return value, called for side effects.

### Examples

```
style()
style(config_file = "tergo.toml", configuration = list())
```

style\_file

*Style a file*

---

**Description**

Style a file

**Usage**

```
style_file(file, configuration = list())
```

**Arguments**

file (character) path to the file to format.

configuration (list) Configuration for formatting. Default list().

**Details**

To see possible configuration options, see [get\\_default\\_config\(\)](#).

**Value**

(logical) whether the file was formatted successfully or skipped. TRUE - formatted successfully, FALSE - skipped.

**Examples**

```
tmp <- tempfile()
file_conn <- file(tmp)
writeLines(c("function(){}", "A<-7"), file_conn)
close(file_conn)
style_file(file = tmp, configuration = list())
unlink(tmp)
```

---

style\_pkg*Style a package*

---

**Description**

Style a package

## Usage

```
style_pkg(  
  path = ".",  
  config_file = "tergo.toml",  
  configuration = list(),  
  force = FALSE,  
  extensions = c(".R", ".r"),  
  verbose = interactive()  
)
```

## Arguments

|               |                                                                                                                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| path          | (character) The path to the package. Default ".".                                                                                                                                                   |
| config_file   | (character) The path to the configuration file. Default "tergo.toml".                                                                                                                               |
| configuration | (list) Configuration for formatting. Default list().                                                                                                                                                |
| force         | (logical(1)) Whether to format the files even if no package was found. TRUE - format the .R and .r files found in the directory (recursive). FALSE exit without formatting anything. Default FALSE. |
| extensions    | (character) The extensions of the files to format. Default c(".R", ".r").                                                                                                                           |
| verbose       | (logical(1)) Whether per file status and run statistics should be printed. Default interactive().                                                                                                   |

## Details

Configuration is read from a file named `tergo.toml` in the root of the package. The precedence of the configuration is (from the highest to lowest):

1. The configuration passed to the function.
2. The configuration file.

To see possible configuration options, see [get\\_default\\_config\(\)](#).

## Value

No return values, called for side effects.

## Examples

```
style_pkg()  
style_pkg(path = "./tergo", config_file = "custom_tergo.toml", verbose = TRUE)
```

---

|            |                   |
|------------|-------------------|
| style_text | <i>Style text</i> |
|------------|-------------------|

---

**Description**

Style text

**Usage**

```
style_text(text, configuration = list())
```

**Arguments**

text (character) the text to style  
configuration (list) Configuration for formatting. Default list().

**Details**

This function is vectorized. To see possible configuration options, see [get\\_default\\_config\(\)](#).

**Value**

(character) The text formatted as R code.

**Examples**

```
code <- "a+b"  
styled <- style_text(code)  
code <- c("a+b", "A<-7")  
styled <- style_text(code)
```

# Index

`get_default_config`, [2](#)  
`get_default_config()`, [3–6](#)

`style`, [3](#)  
`style_file`, [4](#)  
`style_pkg`, [4](#)  
`style_pkg()`, [3](#)  
`style_text`, [6](#)