Package ‘arity’

October 26, 2026

Title Format R Code with Arity

Version 0.2.0

Description Provides R bindings to the deterministic, rule-based formatter from 'arity'. Format R source text or individual files using a style based on the tidyverse R style guide.

License MIT + file LICENSE

URL https://github.com/jolars/arity-r, https://arity.cc

BugReports https://github.com/jolars/arity-r/issues

Depends R (>= 4.2)

Imports rstudioapi (>= 0.5), utils

Suggests testthat (>= 3.2.0), withr

Config/rextendr/version 0.5.0

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

Language en-US

SystemRequirements Cargo (Rust's package manager), rustc >= 1.89.0

NeedsCompilation yes

Author Johan Larsson [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-4029-5945>)

Maintainer Johan Larsson <johan@jolars.co>

Repository CRAN

Date/Publication 2026-10-26 19:50:15 UTC

Contents

format_addin 2
format_text 2

Index **5**

format_addin — *Format R code in RStudio*

Description

Formats selected R code in the source editor, or the whole document when nothing is selected. The addin discovers `arity.toml` from the document's directory, or the working directory for untitled buffers, following the configuration rules of `format_text()`. Syntax verification is enabled.

Usage

```
format_addin()
```

Details

Install arity to make **Format with arity** available in RStudio's Addins menu. You can assign a keyboard shortcut through **Tools > Modify Keyboard Shortcuts**.

Whole-document formatting supports R scripts, `.Rprofile` files, and untitled R buffers. In other documents, such as R Markdown or Quarto files, select complete R expressions in the source editor. Each selection is formatted independently, preserving its initial indentation. If any selection cannot be formatted, the document is left unchanged.

Changes are applied to the editor buffer without saving the file. When formatting a whole document, the cursor is restored to its original row and column, or the closest position if the document becomes shorter.

Value

Invisibly returns `TRUE` when the buffer changed and `FALSE` otherwise.

format_text — *Format R source code*

Description

`format_text()` formats one R source string and returns the result. `format_file()` formats one UTF-8 file in place and reports invisibly whether its contents changed. Both functions use the formatter embedded in the `arity-formatter` Rust crate.

Usage

```
format_text(
  text,
  line_width = NULL,
  indent_width = NULL,
  line_ending = NULL,
  roxygen_markdown = FALSE,
  verify = TRUE,
  config = TRUE
)

format_file(
  path,
  line_width = NULL,
  indent_width = NULL,
  line_ending = NULL,
  roxygen_markdown = FALSE,
  verify = TRUE,
  config = TRUE
)
```

Arguments

`text` A non-missing character scalar containing valid UTF-8 R source code. Strings with a declared encoding are converted to UTF-8.

`line_width` Maximum output line width, from 1 through 1000. `NULL` uses the configuration value, or 80 when unset.

`indent_width` Number of spaces per indentation level, from 1 through 1000. `NULL` uses the configuration value, or 2 when unset.

`line_ending` Output line-ending policy. `"auto"` preserves the style of the first source line ending, `"lf"` and `"crlf"` force a style, and `"native"` uses the current platform's convention. `NULL` uses the configuration value, or `"auto"` when unset.

`roxygen_markdown` Whether roxygen blocks without an explicit `@md` or `@noMd` directive should be parsed in markdown mode.

`verify` Whether to verify syntax preservation, ordinary comment preservation, and idempotence after formatting.

`config` `TRUE` discovers configuration automatically, `FALSE` ignores all configuration files, or a nonempty character scalar names an explicit configuration file. A missing or unreadable selected file is an error.

`path` A non-missing character scalar naming a UTF-8 file. The contents are treated as R source regardless of the file extension.

Details

By default, formatting uses the nearest `arity.toml`, searching upward from the file's directory for `format_file()` or the working directory for `format_text()`. The search includes the repository

root, then stops at a `.git` file or directory, or at the filesystem root. If no project config is found, a nonempty `ARITY_CONFIG` environment variable supplies a fallback config path. Relative explicit and fallback paths use the working directory. Configuration files are not merged. Missing settings use built-in defaults, and explicit formatting arguments override the selected configuration.

The package reads `line-width`, `indent-width`, and `line-ending` from the `[format]` table. Settings for other CLI operations, including exclusions, have no effect. Invalid formatting configuration raises an error before any source is changed.

Value

`format_text()` returns a character scalar. `format_file()` invisibly returns `TRUE` when the file changed and `FALSE` otherwise.

Examples

```
format_text("x<-(1+2)*3^4\n")
path <- tempfile(fileext = ".R")
writeLines("x<-1", path)
format_file(path)
readLines(path)
unlink(path)
```

Index

format_addin, 2
format_file (format_text), 2
format_text, 2
format_text(), *2*