

Package ‘arity’

September 26, 2026

Title Format R Code with Arity

Version 0.1.0

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)

Suggests testthat (>= 3.0.0)

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 2026-09-25 22:30:09 UTC

Contents

format_text	2
Index	4

format_text	<i>Format R source code</i>
-------------	-----------------------------

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; they do not invoke the `arity` CLI or discover an `arity.toml` file.

Usage

```
format_text(
  text,
  line_width = 80L,
  indent_width = 2L,
  line_ending = c("auto", "lf", "crlf", "native"),
  roxygen_markdown = FALSE,
  verify = TRUE
)

format_file(
  path,
  line_width = 80L,
  indent_width = 2L,
  line_ending = c("auto", "lf", "crlf", "native"),
  roxygen_markdown = FALSE,
  verify = TRUE
)
```

Arguments

<code>text</code>	A non-missing character scalar containing valid UTF-8 R source code. Strings with a declared encoding are converted to UTF-8.
<code>line_width</code>	Maximum output line width, from 1 through 1000.
<code>indent_width</code>	Number of spaces per indentation level, from 1 through 1000.
<code>line_ending</code>	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.
<code>roxygen_markdown</code>	Whether roxygen blocks without an explicit <code>@md</code> or <code>@noMd</code> directive should be parsed in markdown mode.
<code>verify</code>	Whether to verify syntax preservation, ordinary comment preservation, and idempotence after formatting.
<code>path</code>	A non-missing character scalar naming a UTF-8 file. The contents are treated as R source regardless of the file extension.

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_file(format_text), 2`
`format_text, 2`