Package: printtree (via r-universe)

June 8, 2026

Type Package

Title Print Directory Trees for R Projects and Folders

Version 0.2.1

Description Quickly visualize 'R' project directory structures with automatic project detection and clean tree output.

License MIT + file LICENSE

URL https://github.com/PrigasG/printtree, https://prigasg.github.io/printtree/

BugReports https://github.com/PrigasG/printtree/issues

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Repository https://prigasg.r-universe.dev

Date/Publication 2026-06-08 15:51:49 UTC

RemoteUrl https://github.com/prigasg/printtree

RemoteRef HEAD

RemoteSha 297c63159186c8d04c41a8cc41e6320ebbc3a614

Contents

print_rtree *Print an R Project or Directory Tree*

Description

Prints a directory tree for a given path. Optionally, detects an RStudio project (`.Rproj`) and can print from a project root.

Usage

```
print_rtree(
  path = NULL,
 ignore = c("renv", ".git", ".Rproj.user", "__pycache__", ".DS_Store", "node_modules",
    ".Rhistory"),
  ignore_type = c("auto", "fixed", "glob", "regex"),
  max_depth = NULL,
  show_hidden = FALSE,
  project = c("auto", "root", "none"),
  search_paths = c(".", "..", "~/Documents", "~/Projects"),
  root_markers = c(".Rproj", "DESCRIPTION"),
  format = c("ascii", "unicode"),
  return_lines = FALSE,
  quiet = FALSE,
  count_footer = TRUE,
  git = FALSE,
  git_legend = TRUE,
  prune = FALSE,
  snapshot = FALSE,
  snapshot_file = "tree.png",
  snapshot_width = 800,
  snapshot_bg = c("white", "black"),
  snapshot_path = "."
)
```

Arguments

`path`	Character. Directory path, project name, or `.Rproj` file. If NULL, uses current directory.
`ignore`	Character vector. Basenames to exclude (e.g., ".git", "renv"). With `ignore_type = "auto"`, entries containing wildcard characters are treated as glob patterns, so values such as `"*.log"` or `"test_*"` work.
`ignore_type`	One of "auto", "fixed", "glob", or "regex". Controls how `ignore` is matched against basenames.
`max_depth`	Integer. Maximum depth to traverse. NULL for unlimited.
`show_hidden`	Logical (TRUE/FALSE). Whether to include hidden files/directories (starting with ".").

project	One of "auto", "root", "none". • "auto": use path as-is (no upward search) • "root": walk upward from path to find a project root (via root_markers) and use it if found • "none": never attempt root detection; print the tree from path
search_paths	Character vector. Used only when path is not an existing directory (treated as a project name). Paths are searched in order.
root_markers	Character vector. Markers used when project = "root" to detect a root directory. Special value ".Rproj" means "any file ending in .Rproj". Common markers include "DESCRIPTION" (R package root) and "_quarto.yml" (Quarto project root).
format	One of "ascii" or "unicode". "ascii" is portable for all terminals.
return_lines	Logical. If TRUE, invisibly return the printed character vector of lines.
quiet	Logical. If TRUE, suppress console output. Useful with return_lines = TRUE.
count_footer	Logical. If TRUE, append a summary like "3 directories, 12 files".
git	Logical. If TRUE, annotate files and directories with porcelain git status markers when path is inside a Git work tree.
git_legend	Logical. If TRUE and git = TRUE, append a short legend explaining the Git status markers.
prune	Logical. If TRUE, omit directories with no displayable children.
snapshot	Logical. If TRUE, gives a visual snapshot of tree.
snapshot_file	Text. Snapshot PNG name if snapshot is set as TRUE.
snapshot_width	Integer. Default set at 800.
snapshot_bg	Either white or black for snapshot background. If white, tree text appears black and vice.
snapshot_path	Character. If snapshot_path is provided, the file is saved there.

Value

Invisible NULL, or a character vector of printed lines if return_lines = TRUE.

Examples

```
# Create a small example directory tree
demo <- file.path(tempdir(), "printtree-demo")
if (dir.exists(demo)) unlink(demo, recursive = TRUE)
dir.create(demo, recursive = TRUE)
dir.create(file.path(demo, "R"))
file.create(file.path(demo, "R", "hello.R"))
file.create(file.path(demo, "README.md"))

# Print the tree
print_rtree(demo)

# Limit depth
```

```
print_rtree(demo, max_depth = 1)

# Save a PNG snapshot to a temporary file
png_file <- tempfile(fileext = ".png")
print_rtree(demo, snapshot = TRUE, snapshot_file = png_file)
```

`write_tree` *Write a Directory Tree to a Text or Markdown File*

Description

Builds a directory tree with the same options as `print_rtree()` and writes it to a plain text or Markdown file.

Usage

```
write_tree(
  path = NULL,
  file,
  format = c("txt", "md"),
  title = NULL,
  create_dirs = TRUE,
  ...
)
```

Arguments

`path`	Character. Directory path, project name, or `.Rproj` file. If NULL, uses current directory.
`file`	Character. Output file path.
`format`	One of "txt" or "md".
`title`	Optional Markdown heading used when `format = "md"`.
`create_dirs`	Logical. If TRUE, create the output file's parent directory when it does not exist.
`...`	Additional arguments passed to `print_rtree()`, such as `ignore`, `max_depth`, `git`, or `prune`.

Value

Invisibly returns the output file path.

Examples

```
demo <- file.path(tempdir(), "printtree-write-demo")
if (dir.exists(demo)) unlink(demo, recursive = TRUE)
dir.create(demo, recursive = TRUE)
file.create(file.path(demo, "README.md"))

out <- tempfile(fileext = ".md")
write_tree(demo, out, format = "md")
```

Index