Package 'ascribe'

August 23, 2026

Title Static Detection and Citation of R Package and Function Usage

Version 0.2.0

Description Scans R source files for package and function use, then builds citations from configurable package universes. Supports .R, .Rmd, and .qmd files, resolves unqualified calls by attachment order and re-export origin, and leaves each package collection to define its own citations. See 'stanflow' <https://github.com/VisruthSK/stanflow> for an example usage.

License MIT + file LICENSE

URL https://ascribe.visruth.com

BugReports https://github.com/VisruthSK/ascribe/issues

Depends R (>= 4.2.0)

Imports brio, cli, fastmatch

Suggests knitr, quarto, spelling, testthat (>= 3.0.0)

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Visruth Srimath Kandali [aut, cre, cph] (ORCID: <https://orcid.org/0009-0005-9097-0688>)

Maintainer Visruth Srimath Kandali <public@visruth.com>

Repository CRAN

Date/Publication 2026-08-22 23:20:43 UTC

Contents

`build_export_index` *Build an inverted export index*

Description

Given a named list mapping package names to character vectors of function names (as produced by `collect_pkg_funs()`), creates an inverted index mapping function names to character vectors of packages that export them.

Usage

```
build_export_index(exports)
```

Arguments

`exports`	Named list. Names are package names, values are character vectors of function names.

Value

Named list mapping function names to character vectors of package names.

Examples

```
exports <- list(
  pkgA = c("foo", "bar"),
  pkgB = c("foo", "baz")
)
build_export_index(exports)
```

build_origin_map *Build an origin map for package functions*

Description

Given a named list mapping package names to character vectors of function names, creates a named character vector mapping `"pkg::fun"` keys to the origin package. Functions whose origin cannot be determined fall back to the providing package.

Usage

```
build_origin_map(exports)
```

Arguments

`exports`	Named list. Names are package names, values are character vectors of function names.

Value

Environment mapping `"pkg::fun"` keys to origin package names.

Examples

```
exports <- list(
  stats = collect_pkg_funs("stats"),
  utils = collect_pkg_funs("utils")
)
build_origin_map(exports)
```

build_universe_data *Build scanner data for a package universe*

Description

Given a character vector of package names, computes the export lists, inverted export index, origin map, and version snapshot needed by `scan_usage()`. All packages must be installed.

Usage

```
build_universe_data(packages)
```

Arguments

`packages`	Character vector of package names.

Value

An `ascribe_universe` object with components:

packages The input package names.

exports Named list mapping package names to character vectors of exported function names (from `collect_pkg_funs()`).

export_index Named list mapping function names to character vectors of packages (from `build_export_index()`).

origin_map Environment mapping `"pkg::fun"` keys to origin packages (from `build_origin_map()`).

pkg_versions Named list mapping package names to version strings.

Examples

```
build_universe_data(c("stats", "utils"))
```

---

cite_usage *Cite package and function use in a project*

Description

Builds citations from `scan_usage()` results. Package collections supply their own citation records and package-citation policy.

Usage

```
cite_usage(
  usage,
  package_citations = new.env(parent = emptyenv(), hash = TRUE),
  function_citations = new.env(parent = emptyenv(), hash = TRUE),
  package_citation = utils::citation,
  always_cite = character(),
  format = c("bibtex", "bibentry")
)
```

Arguments

`usage`	Results returned by `scan_usage()`.
`package_citations`	An environment of package citation entries. Missing packages use `package_citation`.
`function_citations`	An environment of function citation entries, keyed by `"pkg::function"`.
`package_citation`	A function that accepts a package name and returns its citation entries. Defaults to `utils::citation()`.
`always_cite`	Character vector of packages to cite in addition to the packages found by the scan.
`format`	One of `"bibtex"` or `"bibentry"`.

Value

A BibTeX character vector or a bibentry object.

Examples

```
path <- tempfile(fileext = ".R")
writeLines("cli::cli_alert_info('hi'); fastmatch::fmatch(1, 1:5)", path)
universe <- build_universe_data(c("cli", "fastmatch"))
usage <- scan_usage(path, universe)
cite_usage(usage)
unlink(path)
```

---

collect_pkg_funs *Collect exported functions and R6 methods from a package*

Description

Returns a character vector of function names exported by pkg, including methods of exported and namespace-internal R6 classes.

Usage

```
collect_pkg_funs(pkg)
```

Arguments

pkg	Package name (character scalar).

Value

Character vector of function/method names.

Examples

```
collect_pkg_funs("stats")
```

`generate_universe_sysdata` *Generate sysdata.rda for a package universe*

Description

Computes scanner data for the given packages and saves it to `sysdata.rda` with variable names prefixed by `prefix`. This is intended for use in a downstream package's `data-raw/sysdata.R` script.

Usage

```
generate_universe_sysdata(
  packages,
  prefix,
  extra_vars = list(),
  include_scanner_defaults = FALSE,
  file
)
```

Arguments

`packages`	Character vector of package names.
`prefix`	Character scalar used to name the saved objects (e.g., `"stan"` produces `.stan_pkgs`).
`extra_vars`	Named list of additional objects to include in the saved file (e.g., citation environments).
`include_scanner_defaults`	If `TRUE`, also saves `.stdlib_funs` and `.scan_skip_dirs`. Defaults to `FALSE`.
`file`	Output path (required).

Details

The generated variables are:

`.{prefix}_pkgs` Character vector of package names.

`.{prefix}_exports` Named list of exported functions per package.

`.{prefix}_export_index` Inverted index: function name to packages.

`.{prefix}_origin_map` Environment: `"pkg::fun"` keys to origin.

`.{prefix}_pkg_versions` Named list of version strings.

When `include_scanner_defaults` is `TRUE`, `.stdlib_funs` and `.scan_skip_dirs` are also saved.

Value

Invisibly returns the result of `build_universe_data()`.

Examples

```
file <- tempfile(fileext = ".rda")
generate_universe_sysdata(c("stats", "utils"), "my", file = file)
unlink(file)
```

---

resolve_origin *Resolve the origin package of an exported function*

Description

Given a package and function name, determines which package the function actually originates from (handling re-exports).

Usage

```
resolve_origin(pkg, name)
```

Arguments

pkg	Package name (character scalar).
name	Function name (character scalar).

Value

The origin package name, or `NA_character_` if undetermined.

Examples

```
resolve_origin("stats", "median")
```

---

scan_usage *Find used functions and packages*

Description

Statically scans R source files for package attachments and function calls. It recognizes `library()`, `require()`, `requireNamespace()`, and `use()`.

Usage

```
scan_usage(
  path = ".",
  universe,
  ignore_unqualified_functions = .stdlib_funs,
  strict = FALSE,
  skip_dirs = .scan_skip_dirs,
  metapackages = NULL,
  use_knitr = FALSE
)
```

Arguments

path	A single project directory (searched recursively) or a vector of files (.R/.Rmd/.qmd).
universe	A universe returned by build_universe_data().
ignore_unqualified_functions	Defaults to exports from base R packages listed in stdlib_funs(). Character vector of function names to ignore when attributing (unqualified) calls. Calls like pkg::fun() will NOT be ignored even if fun is in ignore_unqualified_functions, since they are namespaced.
strict	If FALSE (default), warn on ambiguous function calls whose origin cannot be determined exactly. If TRUE, abort on ambiguous calls.
skip_dirs	Character vector of directory names to skip when scanning a directory. Defaults to scan_skip_dirs().
metapackages	Named list mapping attached package names to additional packages that should be treated as co-attached for unqualified resolution. Defaults to NULL.
use_knitr	Logical. If TRUE, parse .Rmd and .qmd files with knitr::purl(), which resolves knitr features the in-house parser ignores, such as child documents. It also comments out eval=FALSE and purl=FALSE chunks, so usage in them goes unrecorded. Defaults to FALSE.

Details

Explicit package references from library(), require(), requireNamespace(), use(), and pkg::fun are only recorded when their package is included in allowed_packages. The scanner attributes an unqualified function only when library() or require() attached a package earlier in the same file and the supplied indexes can resolve the call. metapackages can add packages to that attachment set. If several attached packages export the function, the most recently attached match wins. The scanner attributes known re-exports to their origin package and otherwise to the resolved package.

Value

A list of packages, resolved functions, and ambiguous function calls.

Examples

```
path <- tempfile(fileext = ".R")
writeLines(
  c(
    "# one messy analysis file",
    "library(stats)",
    "requireNamespace(\"utils\")",
    "filter(1:10, rep(1, 3))",
    "utils::head(letters)"
  ),
  path
)
universe <- build_universe_data(c("stats", "utils"))
scan_usage(path, universe, ignore_unqualified_functions = character())
unlink(path)
```

Index