

Package: ascribe (via r-universe)

July 23, 2026

Title Static Detection and Citation of R Package and Function Usage

Version 0.1.0

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 cli, fastmatch

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

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Repository <https://r-multiverse-staging.r-universe.dev>

Date/Publication 2026-07-23 06:30:41 UTC

RemoteUrl <https://github.com/VisruthSK/ascribe>

RemoteRef v0.1.0

RemoteSha 1bfd635960589d7f58c988cbe0d8300cc968b3e8

Contents

build_export_index	2
build_origin_map	3
build_universe_data	3
cite_usage	4
collect_pkg_funs	5
generate_universe_sysdata	6
resolve_origin	7
scan_usage	7
Index	10

build_export_index	<i>Build an inverted export index</i>
--------------------	---------------------------------------

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	<i>Build an origin map for package functions</i>
------------------	--

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

Named character vector mapping "pkg: : fun" to origin package.

Examples

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

build_universe_data	<i>Build scanner data for a package universe</i>
---------------------	--

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

A named list 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 Named character vector 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	<i>Cite package and function use in a project</i>
------------	---

Description

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

Usage

```
cite_usage(
  usage,
  package_citations = list(),
  function_citations = list(),
  package_citation = utils::citation,
  always_cite = character(),
  format = c("bibtex", "bibentry")
)
```

Arguments

usage	Results returned by <code>scan_usage()</code> .
package_citations	A named list or environment of package citation entries. Missing packages use <code>package_citation</code> .
function_citations	A named list or 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 <code>utils::citation()</code> .
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("stats::median(1:3)", path)
universe <- build_universe_data(c("stats", "tools"))
usage <- scan_usage(path, universe$packages, universe$export_index, universe$origin_map)
cite_usage(usage)
unlink(path)
```

collect_pkg_funs	<i>Collect exported functions and R6 methods from a package</i>
------------------	---

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 = "R/sysdata.rda"
)
```

Arguments

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

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` Named character vector: "pkg::fun" 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	<i>Resolve the origin package of an exported function</i>
----------------	---

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	<i>Find used functions and packages</i>
------------	---

Description

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

Usage

```
scan_usage(
  path = ".",
  allowed_packages,
  export_index,
  origin_map,
  ignore_unqualified_functions = .stdlib_funs,
  strict = FALSE,
  skip_dirs = .scan_skip_dirs,
  metapackages = NULL,
  use_knitr = FALSE,
  quiet = FALSE
)
```

Arguments

<code>path</code>	A single project directory (searched recursively) or a vector of files (.R/.Rmd/.qmd).
<code>allowed_packages</code>	Character vector of package namespaces to attribute calls to.
<code>export_index</code>	Named list mapping function names to packages.
<code>origin_map</code>	Named character vector mapping <code>pkg::fun</code> keys to the origin package.
<code>ignore_unqualified_functions</code>	Defaults to exports from base R packages listed in <code>stdlib_funs()</code> . Character vector of function names to ignore when attributing (unqualified) calls. Calls like <code>pkg::fun()</code> will NOT be ignored even if <code>fun</code> is in <code>ignore_unqualified_functions</code> , since they are namespaced.
<code>strict</code>	If FALSE (default), warn on ambiguous function calls whose origin cannot be determined exactly. If TRUE, abort on ambiguous calls.
<code>skip_dirs</code>	Character vector of directory names to skip when scanning a directory. Defaults to <code>scan_skip_dirs()</code> .
<code>metapackages</code>	Named list mapping attached package names to additional packages that should be treated as co-attached for unqualified resolution. Defaults to NULL.
<code>use_knitr</code>	Logical. If TRUE, parse .Rmd and .qmd files with <code>knitr::purl()</code> . This is more accurate for knitr/quarto chunk handling but much slower than the default in-house parser. Defaults to FALSE.
<code>quiet</code>	Logical. If TRUE, suppresses status messages. 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
)
scan_usage(
  path,
  allowed_packages = c("stats", "utils"),
  export_index = list(filter = "stats"),
  origin_map = c("stats::filter" = "stats"),
  ignore_unqualified_functions = character(),
  quiet = TRUE
)
unlink(path)
```

Index

`build_export_index`, [2](#)
`build_export_index()`, [4](#)
`build_origin_map`, [3](#)
`build_origin_map()`, [4](#)
`build_universe_data`, [3](#)
`build_universe_data()`, [6](#)

`cite_usage`, [4](#)
`collect_pkg_funs`, [5](#)
`collect_pkg_funs()`, [2](#), [4](#)

`generate_universe_sysdata`, [6](#)

`resolve_origin`, [7](#)

`scan_usage`, [7](#)
`scan_usage()`, [3](#), [4](#)

`utils::citation()`, [4](#)