Package ‘covtracer’

July 8, 2024

Title  Contextualizing Tests
Version  0.0.1
Description  Dissects a package environment or ‘covr’ coverage object in order to cross
reference tested code with the lines that are evaluated, as well as linking
those evaluated lines to the documentation that they are described within.
Connecting these three pieces of information provides a mechanism of
linking tests to documented behaviors.

URL  https://github.com/genentech/covtracer
BugReports  https://github.com/genentech/covtracer/issues
Depends  R (>= 3.2.0)
Imports  tools, stats, methods
Suggests  testthat, covr (>= 3.5.2), withr, R6, cli, dplyr, igraph,
  knitr, rmarkdown
License  MIT + file LICENSE
Encoding  UTF-8
RoxygenNote  7.3.1
VignetteBuilder  knitr
NeedsCompilation  no
Author  Doug Kelkhoff [cre, aut] (<https://orcid.org/0009-0003-7845-4061>),
  Szymon Maksymiuk [aut] (<https://orcid.org/0000-0002-3120-1601>),
  Andrew McNeil [aut],
  F. Hoffmann-La Roche AG [cph, fnd]
Maintainer  Doug Kelkhoff <doug.kelkhoff@gmail.com>
Repository  CRAN
Date/Publication  2024-07-08 05:00:07 UTC

Contents

  as.data.frame.list_of_srcref  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  as.package  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1
as.data.frame.list_of_srcref

Coerce a list_of_srcref object to a data.frame

Description

Coerce a list_of_srcref object to a data.frame
as.data.frame.list_of_srcref

Usage

## S3 method for class 'list_of_srcref'
as.data.frame(
  x,
  ...,  
  use.names = TRUE,
  expand.srcref = FALSE,
  row.names = NULL
)

Arguments

x  A list_of_srcref object

...  Additional arguments unused

use.names  A logical indicating whether the names of x should be used to create a name column.

expand.srcref  A logical indicating whether to expand the components of srcref objects into separate columns.

row.names  NULL or a single integer or character string specifying a column to be used as row names, or a character or integer vector giving the row names for the data frame.

Value

A data.frame with one record per srcref and variables:

name  Names of the srcref objects, passed using the names of x if use.names = TRUE

srcref  srcref objects if expand.srcrefs = FALSE

srcref, line1, byte1, line2, col1, col2, parsed1, parsed2  The srcref file location if it can be determined. If an absolute path can’t be found, only the base file name provided in the srcref object and the numeric components of the srcref objects if expand.srcrefs = TRUE

Examples

pkg <- system.file("examplepkg", package = "covtracer")
install.packages(
  pkg,
  type = "source",
  repos = NULL,
  quiet = TRUE,
  INSTALL_opts = "--with-keep.source"
)

as.data.frame(pkg_srcrefs("examplepkg"))
as.package  

A simple alternative to devtools::as.package

Description

Functionally identical to devtools' as.package, but without interactive options for package creation.

Usage

as.package(x)

Arguments

x  
A package object to coerce

Value

A package object

Note

Code inspired by devtools load_pkg_description with very minor edits to further reduce devtools dependencies.

as_list_of_srcref

Create an S3 list of srcref objects

Description

Create an S3 list of srcref objects

Usage

as_list_of_srcref(x)

## S3 method for class 'environment'
as_list_of_srcref(x)

## S3 method for class 'list'
as_list_of_srcref(x)

Arguments

x  
A list or single srcref to coerce to a list_of_srcref
as_test_desc

Value
A list_of_src_ref class object

Description
Wrap object in test description derivation data
Adds "testthat" style

Usage
as_test_desc(x, type = "call")
as_testthat_desc(x)

Arguments
x A test description string to bind style data to
type A type class to attribute to the test description. Defaults to "call".

Value
A test_description subclass object with additional style attribute indicating how the test description was derived.

coverage_check_has_recorded_tests

Description
Check that the coverage object retains testing information

Usage
coverage_check_has_recorded_tests(coverage, warn = TRUE)

Arguments
coverage a covr coverage object
warn Whether to warn when it is uncertain whether the tests were recorded. It may be uncertain if tests were recorded if there are no tested R code traces.
Value

Used for side-effects of emitting an error when a coverage object does not contain recorded traces, or a warning when a coverage object appears to have no tests.

See Also

Other coverage_tests: `coverage_get_tests()`, `coverage_has_recorded_tests()`

---

**coverage_get_tests**

*Retrieve test traces from a coverage object*

Description

Assumes the coverage object was produced while `option(covr.record_tests = TRUE)`.

Usage

`coverage_get_tests(coverage)`

Arguments

- `coverage` a `covr` coverage object

Value

A list of tests evaluated when using `covr`

See Also

Other coverage_tests: `coverage_check_has_recorded_tests()`, `coverage_has_recorded_tests()`

---

**coverage_has_recorded_tests**

*Test that the coverage object retains testing information*

Description

Test whether the coverage object has expected fields produced when coverage was captured with `option(covr.record_tests = TRUE)`.

Usage

`coverage_has_recorded_tests(coverage)`
**expr_str**

**Arguments**

coverage  
a covr coverage object

**Value**

A logical value, indicating whether the coverage object has recorded tests, or NA when it does not appear to have traced any test code.

**See Also**

Other coverage_tests: `coverage_check_has_recorded_tests()`, `coverage_get_tests()`

---

**expr_str**

*Convert an expression, call or symbol to a single-line string*

**Description**

Convert an expression, call or symbol to a single-line string

**Usage**

`expr_str(ref)`

**Arguments**

ref  
a srcref

**Value**

The given expression, formatted as a string with prefixes for symbols and generics.

---

**flat_map_srcrefs**

*Map srcrefs over an iterable object, Filtering non-srcref results*

**Description**

Map srcrefs over an iterable object, Filtering non-srcref results

**Usage**

`flat_map_srcrefs(xs, ns = NULL, breadcrumbs = character())`
format.list_of_srcref

Arguments

xs  Any iterable object
ns  A character namespace name to attribute to objects in xs. If xs objects themselves have namespaces attributed already to them, the namespace will not be replaced.
breadcrumbs Recursive methods are expected to propagate a vector of "breadcrumbs" (a character vector of namespace names encountered while traversing the namespace used as a memory of what we’ve seen already), which is used for short-circuiting recursive environment traversal.

Value

A list of srcrefs

Description

Format list_of_srcref as character

Usage

## S3 method for class 'list_of_srcref'
format(x, ..., full.names = FALSE, full.num = FALSE)

Arguments

x  A list_of_srcref object
... Additional arguments unused
full.names  A logical value indicating whether to use full file paths when formatting srcrefs.
full.num  A logical value indicating whether to use all numeric srcref components when formatting srcrefs.

Value

A character vector of formatted strings
getSrcFilepath

Get the full path to the srcref file

Description

Get the full path to the srcref file

Usage

getSrcFilepath(x)

Arguments

x A srcref or list_of_srcref object

Value

A character vector of source file paths.

get_namespace_object_names

Get namespace exports, filtering methods tables and definitions

Description

Get namespace exports, filtering methods tables and definitions

Usage

get_namespace_object_names(ns)

Arguments

ns A namespace object

Value

The names of exported objects, filtering internal method tables and metadata.
is_srcref

Test whether an object is a srcref object

Description

Test whether an object is a srcref object

Usage

is_srcref(x)

Arguments

x

Any object

Value

A logical indicating whether object is a srcref

join_on_containing_srcrefs

Join srcref data.frames by intersection of srcref spans

Description

References to source code are defined by the source code line and column span of the relevant source code. This function takes data frames containing that information to pair source code in one data frame to source code from another. In this case, source code from the left hand data frame is paired if it is entirely contained within a record of source code in the right hand data frame.

Usage

join_on_containing_srcrefs(x, y, by = c(srcref = "srcref"))

Arguments

x

A data.frame, as produced by `as.data.frame` applied to a `list_of_srcref`, against which y should be joined.

y

A data.frame, as produced by `as.data.frame` applied to a `list_of_srcref`, joining data from srcrefs data which encompasses srcrefs from x.

by

A named character vector of column names to use for the merge. The name should be the name of the column from the left data.frame containing a `list_of_srcref` column, and the value should be the name of a column from the right data.frame containing a `list_of_srcref` column.
**match_containing_srcrefs**

*Match srcrefs against srcrefs that contain them*

**Description**

Provided two lists of srcref objects, find the first srcrefs in `r` that entirely encapsulate each respective srcref in `l`, returning a list of indices of srcrefs in `r` for each srcref in `l`.

**Usage**

```
match_containing_srcrefs(l, r)
```

**Arguments**

- `l` | A list of srcref object
- `r` | A list of srcref object

**Value**

A integer vector of the first index in `r` that fully encapsulate the respective element in `l`.

---

**new_empty_test_trace_tally**

*Build an empty covr-style test trace mapping*

**Description**

Build an empty covr-style test trace mapping

**Usage**

```
new_empty_test_trace_tally()
```

**Value**

An empty test-trace matrix, as provided by covr
**obj_namespace_name**  
*Get namespace export namespace name*

**Description**

For most objects, this will be identical to the namespace name provided, but reexports will retain their originating package’s namespace name. This helper function helps to expose this name to determine which exports are reexports.

**Usage**

```
obj_namespace_name(x, ns)
```

**Arguments**

- **x**  
  A value to find within namespace `ns`

- **ns**  
  A package namespace

**Value**

A character string representing a namespace or similar

---

**package_check_has_keep_source**  
*Verify that the package collection contains srcref information*

**Description**

Test whether the package object collection contains srcref attributes.

**Usage**

```
package_check_has_keep_source(env)
```

**Arguments**

- **env**  
  A package namespace environment or iterable collection of package objects

**Value**

Used for side effect of throwing an error when a package was not installed with srcrefs.
pkg_srcrefs

Extract all srcref objects of objects within a package namespace

Description

Extract all the srcref objects of objects within a package namespace

Usage

pkg_srcrefs(x)

## S3 method for class 'environment'
pkg_srcrefs(x)

## S3 method for class 'character'
pkg_srcrefs(x)

## S3 method for class 'coverage'
pkg_srcrefs(x)

Arguments

x A package_coverage coverage object, from which the name of the package used is extracted.

Value

A list_of_srcref

See Also

as.data.frame.list_of_srcref

Other srcrefs: test_srcrefs(), trace_srcrefs()

Examples

pkg <- system.file("examplepkg", package = "covtracer")
install.packages(
  pkg,
  type = "source",
  repos = NULL,
  quiet = TRUE,
  INSTALL_opts = "--with-keep.source"
)
pkg_srcrefs("examplepkg")
pkg_srcrefs_df

Create a data.frame of package srcref objects

Description

Create a data.frame of package srcref objects

Usage

pkg_srcrefs_df(x)

Arguments

x  A package_coverage coverage object, from which the name of the package used is extracted.

Value

A data.frame with a record for each source code block with variables:

- name  A character Rd alias for the package object
- srcref  The srcref of the associated package source code

See Also

srcrefs test_trace_mapping

Other srcrefs_df: test_srcrefs_df(), trace_srcrefs_df()

Examples

pkg <- system.file("examplepkg", package = "covtracer")
install.packages(
  pkg,
  type = "source",
  repos = NULL,
  quiet = TRUE,
  INSTALL_opts = "--with-keep.source"
)
pkg_srcrefs_df("examplepkg")
Create a tabular representation of man file information

Description

Provides Rd index info with a few additional columns of information about each exported object. Returns one record per documented object, even if multiple objects alias to the same documentation file.

Usage

Rd_df(x)

Arguments

x A package object to coerce

Value

A data.frame of documented object information with variables:

- **index** A numeric index of documentation files associated with documentation objects
- **file** A character filename of the Rd file in the "man" directory
- **filepath** A character file path of the Rd file in the "man" directory
- **alias** character object names which are aliases for the documentation in filepath
- **is_exported** A logical indicator of whether the aliased object is exported from the package namespace
- **doctype** A character representing the Rd docType field.

Examples

```r
package_source_dir <- system.file("examplepkg", package = "covtracer")
Rd_df(package_source_dir)
```

Retrieve srcrefs

Description

This function takes a code collection and returns a list of related srcref objects with list names that associate the srcref with a name or alias that could be used to find documentation. Code collections include structures such as package namespaces, environments, function definitions, methods tables or class generators - any object which enapsulates a single or set of srcref objects.
Usage

srcrefs(x, ...)

## Default S3 method:
srcrefs(x, ..., srcref_names = NULL, breadcrumbs = character())

## S3 method for class 'list'
srcrefs(x, ..., srcref_names = NULL, breadcrumbs = character())

## S3 method for class 'namespace'
srcrefs(x, ..., breadcrumbs = character())

## S3 method for class 'environment'
srcrefs(x, ..., breadcrumbs = character())

## S3 method for class 'R6ClassGenerator'
srcrefs(x, ..., srcref_names = NULL, breadcrumbs = character())

## S3 method for class 'standardGeneric'
srcrefs(x, ..., srcref_names = NULL)

## S3 method for class 'nonstandardGenericFunction'
srcrefs(x, ..., srcref_names = NULL)

## S3 method for class 'MethodDefinition'
srcrefs(x, ..., srcref_names = NULL)

Arguments

x  An object to source srcrefs from
...

srcref_names  An optional field used to supercede any discovered object names when choosing which names to provide in the returned list.
breadcrumbs  Recursive methods are expected to propegate a vector of "breadcrumbs" (a character vector of namespace names encountered while traversing the namespace used as a memory of what we’ve seen already), which is used for short-circuiting recursive environment traversal.

Details

For most objects, this is a one-to-one mapping of exported object names to their srcref, just like you would get using getNamespace(). However, for classes and methods, this can be a one-to-many mapping of related documentation to the multiple srcrefs that are described there. This is the case for S3 generics, S4 objects and R6 objects.

Objects without any related srcrefs, such as any datasets or objects created at package build time will be omitted from the results.
srcref_expr

Value

A list of srcref objects. Often, has a length of 1, but can be larger for things like environments, namespaces or generic methods. The names of the list reflect the name of the Rd name or alias that could be used to find information related to each srcref. Elements of the list will have attribute "namespace" denoting the source environment namespace if one can be determined for the srcref object.

Examples

# examples use `with` to execute within namespace as function isn't exported
ns <- getNamespace("covtracer")

# load and extract srcrefs for a package
with(ns, srcrefs(getNamespace("covtracer")))

# extract srcrefs for functions
with(ns, srcrefs(srcrefs))

srcref_expr

Parse the expression associated with a srcref

Description

Parse the expression associated with a srcref

Usage

srcref_expr(ref)

Arguments

ref a srcref

Value

A parsed srcref object
**srcref_str**

*Convert a srcref into a string*

**Description**

Convert a srcref into a string

**Usage**

```
srcref_str(ref)
```

**Arguments**

- `ref` a srcref

**Value**

A string representing the srcref

---

**test_description**

*Parse a test description from the calling expression*

**Description**

In the general case, a simple indicator of the source file and line number is used as a test description. There are some special cases where more descriptive information can be extracted:

**Usage**

```
test_description(x)
```

**Arguments**

- `x` a unit test call stack or expression.

**Details**

- `testthat` If the test used `test_that`, then the description (desc parameter) is extracted and evaluated if need be to produce a descriptive string. Nested calls to `test_that` currently return the outermost test description, although this behavior is subject to change.

**Value**

A string that describes the test. If possible, this will be a written description of the test, but will fall back to the test call as a string in cases where no written description can be determined.
test_description_test_that

Parse the test description from a test_that call

Description
Parse the test description from a test_that call

Usage
test_description_test_that(x, ...)

Arguments
x A test_that call object
... Additional arguments unused

Value
A character description, parsed from a test_that::test_that call

test_description_test_that_describe

Parse the test description from a describe call

Description
Parse the test description from a describe call

Usage
test_description_test_that_describe(x, ...)

Arguments
x A test_that::describe call object
... Additional arguments unused

Value
A character description, parsed from a test_that::describe call
test_description_test_that_describe_it

*Parse the test description from a it call*

**Description**

Parse the test description from a it call

**Usage**

```
test_description_test_that_describe_it(x, ...)
```

**Arguments**

- `x`: A `test_that::it` call object
- `...`: Additional arguments unused

**Value**

A character description, parsed from a `test_that::it` call

---

test_screfs

*Extract test srcref objects*

**Description**

Extract test srcref objects

**Usage**

```
test_screfs(x)
```

- **S3 method for class 'coverage'**

```
test_screfs(x)
```

**Arguments**

- `x`: A `package_coverage` coverage object, from which the test srcrefs are extracted.

**Value**

A `list_of_srcref`
See Also

as.data.frame.list_of_srcref
Other srcrefs: pkg_srcrefs(), trace_srcrefs()

Examples

options(covr.record_tests = TRUE)
pkg_path <- system.file("examplepkg", package = "covtracer")
cov <- covr::package_coverage(pkg_path)
test_srcrefs(cov)

test_srcrefs_df Create a data.frame of coverage test srcref objects

Description

Extract unit test srcrefs from a coverage object. A test name will be derived from the test source code, preferably from a written annotation, but otherwise falling back to using a code snippet. srcrefs are unique for each expression executed within a testing suite.

Usage

test_srcrefs_df(x)

Arguments

x A package_coverage coverage object, from which the name of the package used is extracted.

Value

A data.frame of test srcrefs extracted from a coverage object. Contains one record for each srcref with variables:

name A character test description. For testthat tests, the desc parameter will be used, otherwise a snippet of code will be used for the test name
srcref A srcref object describing the location of the test
test_type A character indicating the structure of the test. One of "testthat", "call" or NULL

See Also

srcrefs test_trace_mapping
Other srcrefs_df: pkg_srcrefs_df(), trace_srcrefs_df()
test_trace_df

Build a traceability matrix that links documented behaviors to unit tests

Description

Intercept unit test coverage reports and process results to link evaluated functions to the unit tests which trigger their evaluation. In doing so, we can then link the associated function documentation of each object to the tests that triggered their evaluation as a way of reusing existing documentation to generate specifications.

Usage

test_trace_df(x, ...)

## S3 method for class 'coverage'
test_trace_df(
  x,
  ...
)

Arguments

x A package object, name, source code path or coverage result to use as the bases of tracing tests. Coverage results must have been produced using options(covr.record_tests = TRUE).

... Additional arguments unused

pkg A package object as produced by as.package, if a specific package object is to be used for inspecting the package namespace.

aggregate_by NULL or a function by which to aggregate recurring hits counts and direct columns from a test to a trace. If NULL, no aggregation will be applied. (Default sum)

Value

A data.frame of tests and corresponding traces
Create a data.frame mapping tests to coverage traces

Description

Extract a matrix used to relate test code to the traces that each test evaluates.

Usage

test_trace_mapping(x)

Arguments

x A coverage object produced with options(covr.record_tests = TRUE).

Value

A data.frame with one record for each line of code executed, with variables:

- **test** The index of the test that was executed, reflecting the order in which tests are executed
- **depth** The call stack depth when the coverage trace was evaluated
- **i** The index of the expression evaluated by each test. This can be used to recover an order of trace execution for a given test index
- **trace** The index of the coverage trace that was evaluated

See Also

srcrefs_df srcrefs

Examples

```r
go
options(covr.record_tests = TRUE)
pkg_path <- system.file("examplepkg", package = "covtracer")
cov <- covr::package_coverage(pkg_path)
test_trace_mapping(cov)
```
trace_srcrefs

**Extract srcref objects from coverage object traces**

**Description**

Extract srcref objects from coverage object traces

**Usage**

```r
trace_srcrefs(x)
```

**Arguments**

- `x` (link[covr]{package_coverage}) A covr coverage object produced with options(covr.record_tests = TRUE).

**Value**

A `list_of_srcref`

**See Also**

as.data.frame.list_of_srcref

Other srcrefs: pkg_srcrefs(), test_srcrefs()

---

trace_srcrefs_df

**Create a data.frame of coverage trace srcref objects**

**Description**

Extract coverage traces. Traces are the traced lines of code counted when evaluating code coverage, which are used for counting expression evaluation. Each traced is a unique expression within a package's source code.

**Usage**

```r
trace_srcrefs_df(x)
```

**Arguments**

- `x` A package_coverage coverage object, from which the name of the package used is extracted.
with_pseudosrcref

Value

A data.frame, where each record is a trace srcref with variables:

- **name**: A character identifier. This will use the names of the elements of a coverage object, which are srcref "keys".
- **srcref**: A srcref object of the trace source code location

See Also

srcrefs test_trace_mapping

Other srcrefs_df: pkg_srcrefs_df(), test_srcrefs_df()

Examples

options(covr.record_tests = TRUE)
pkg_path <- system.file("examplepkg", package = "covtracer")

cov <- covr::package_coverage(pkg_path)
trace_srcrefs_df(cov)

with_pseudo_srcref

For consistency, stub calls with srcref-like attributes

Description

Most relevant data can be traced to an existing srcref. However, some data, such as test traces from coverage objects, are likely cleaned up and their srcfiles deleted, causing a barrage of warnings any time these objects are printed.

Usage

with_pseudo_srcref(call, file, lloc)

Arguments

- **call**: Any code object, most often a call object
- **file**: A filepath to bind as a srcfile object
- **lloc**: A srcref-like lloc numeric vector

Details

A pseudo_srcref adds in the srcref data but continues to preserve the expression content. This allows these expression objects to be pretty-printed like srcrefs when included as a list_of_srcref data.frame column.

Value

A with_pseudo_srcref object, mimicking the structure of srcref
Index

* **coverage_tests**
  - coverage_check_has_recorded_tests, 5
  - coverage_get_tests, 6
  - coverage_has_recorded_tests, 6

* **srcrefs_df**
  - pkg_srcrefs_df, 14
  - test_srcrefs_df, 21
  - trace_srcrefs_df, 24

* **srcrefs**
  - pkg_srcrefs, 13
  - test_srcrefs, 20
  - trace_srcrefs, 24

  - as.data.frame.list_of_srcref, 2
  - as.package, 4
  - as_list_of_srcref, 4
  - as_test_desc, 5
  - as_testthat_desc (as_test_desc), 5
  - coverage, 21, 24, 25
  - coverage_check_has_recorded_tests, 5, 6, 7
  - coverage_get_tests, 6, 6, 7
  - coverage_has_recorded_tests, 6, 6
  - covr, 5–7, 24
  - expr_str, 7
  - flat_map_srcrefs, 7
  - format.list_of_srcref, 8
  - get_namespace_object_names, 9
  - getSrcFilepath, 9
  - is_srcref, 10
  - join_on_containing_srcrefs, 10
  - match_containing_srcrefs, 11
  - new_empty_test_trace_tally, 11
  - obj_namespace_name, 12
  - package_check_has_keep_source, 12
  - package_coverage, 13, 14, 20, 21, 24
  - pkg_srcrefs, 13, 21, 24
  - pkg_srcrefs_df, 14, 21, 25
  - Rd_df, 15
  - srcref_expr, 17
  - srcref_str, 18
  - srcrefs, 15
  - test_description, 18
  - test_description_test_that, 19
  - test_description_test_that_describe, 19
  - test_description_test_that_describe_it, 20
  - test_srcrefs, 13, 20, 24
  - test_srcrefs_df, 14, 21, 25
  - test_that, 18
  - test_trace_df, 22
  - test_trace_mapping, 23
  - trace_srcrefs, 13, 21, 24
  - trace_srcrefs_df, 14, 21, 24
  - with_pseudo_srcref, 25