Package 'biocroxytest'

May 29, 2024

Title Handle Long Tests in Bioconductor Packages

Version 1.0.0

Description This package provides a roclet for roxygen2 that identifies and processes code blocks in your documentation marked with `@longtests`. These blocks should contain tests that take a long time to run and thus cannot be included in the regular test suite of the package. When you run `roxygen2::roxygenise` with the `longtests_roclet`, it will extract these long tests from your documentation and save them in a separate directory. This allows you to run these long tests separately from the rest of your tests, for example, on a continuous integration server that is set up to run long tests.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE, roclets = c(``rd", ``collate", ``namespace"))

RoxygenNote 7.3.0

URL https://github.com/xec-cm/biocroxytest

BugReports https://github.com/xec-cm/biocroxytest/issues

biocViews Software, Infrastructure

Imports cli, glue, roxygen2, stringr

Suggests BiocStyle, here, knitr, rmarkdown, testthat (>= 3.0.0)

Depends R (>= 4.4.0)

Config/testthat/edition 3

Config/testthat/parallel false

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/biocroxytest

git_branch RELEASE_3_19

git_last_commit c8048b1

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29

Author Francesc Catala-Moll [aut, cre]
(<https://orcid.org/0000-0002-2354-8648>)

Maintainer Francesc Catala-Moll <fcatala@irsicaixa.es>

Contents

`block_to_testthat` *Convert a roxygen2 block to a testthat file*

Description

This is an internal function that converts a roxygen2 block to a testthat file. It first checks if the block has any 'longtests' tags. If not, it returns NULL. Then, it extracts the 'longtests' and 'examples' tags from the block and assigns them to the 'longtests' and 'examples' fields of the testthat file, respectively. It also assigns the basename of the block's file to the 'filename' field, and the alias of the block's object to the 'functionname' field. If the block's line is not NULL, it appends it to the 'functionname' field.

Usage

```
block_to_testthat(block)
```

Arguments

`block`	A roxygen2 block.

Value

A list representing a testthat file, or NULL if the block has no 'longtests' tags.

`internal_longtests_roclet_clean` *Clean up longtest files generated by biocroxytest*

Description

This is an internal function that cleans up test files generated by biocroxytest. It checks each file in the provided list of test files. If a file was generated by biocroxytest, it will be removed. If a file was not generated by biocroxytest, a message will be printed to inform the user, and the file will not be removed.

Usage

```
internal_longtests_roclet_clean(testfiles)
```

Arguments

`testfiles`	A character vector representing the paths of the test files to check and clean.

Value

invisible

`internal_longtests_roclet_output` *Write long test results to files*

Description

This function takes three arguments: results, base_path, and prefix. The function iterates over the elements of results and writes the content of each element to a file in the directory specified by base_path and prefix. If the file was not created by biocroxytest, the function informs the user that some files were not removed.

Usage

```
internal_longtests_roclet_output(
  results,
  base_path,
  prefix = "test-biocroxytest"
)
```

Arguments

`results`	A list of results.
`base_path`	The base path where the files will be written.
`prefix`	The prefix that will be used to name the files.

Value

A list of file paths.

`internal_longtests_roclet_process` *Process blocks for 'longtests' roclet*

Description

This is an internal function that processes a list of blocks for the 'longtests' roclet. It first collects and annotates the 'rdname' of the blocks. Then, it loops over each block and converts it to a testthat file using the 'block_to_testthat' function. If the resulting testthat file has a filename and contains 'longtests', it is added to the list of test files. Finally, it processes the test files using the 'process_testfiles' function.

Usage

```
internal_longtests_roclet_process(blocks, ...)
```

Arguments

`blocks`	A list of roxygen2 blocks.
`...`	Additional arguments.

Value

A list with a single element 'longtests', which contains the processed test files.

`longtests_roclet` *Roclet for Long Tests*

Description

This roclet is used to identify and process code blocks in your documentation that are marked with `@longtests`. The `longtests_roclet` function creates a new roclet, which is a plugin to the `roxygen2` package. Roclets are responsible for parsing the R scripts of a package and producing the relevant documentation files.

Usage

```
longtests_roclet()
```

Details

The `longtests_roclet` specifically looks for code blocks in your documentation that are annotated with the `@longtests` tag. These code blocks should contain tests that take a long time to run, and thus cannot be included in the regular test suite of the package.

When you run `roxygen2::roxygenise` with the `longtests_roclet`, it will extract these long tests from your documentation and save them in a separate directory. This allows you to run these long tests separately from the rest of your tests, for example, on a continuous integration server that is set up to run long tests.

Value

A roclet that can be used with `roxygen2` to process code blocks marked with `@longtests`. This roclet will produce a set of test files in a separate directory, which can be run independently of the rest of your test suite.

Examples

```
# Create a new roclet
longtests_roclet()
```

`made_by_biocroxytest` *Check if a file is generated by biocroxytest*

Description

This is an internal function that checks if a file at a given path is generated by biocroxytest. It checks if the file exists, if the name matches the pattern "test-biocroxytest-.*\.R$", and if the header of the file starts with "Generated by biocroxytest". If the file exists and the name matches the pattern but the header does not start with "Generated by biocroxytest", the function will return FALSE.

Usage

```
made_by_biocroxytest(path)
```

Arguments

`path`	A character string representing the path of the file to check.

Value

TRUE if all checks pass, FALSE otherwise.

`process_testfiles` *Process test files for 'longtests'*

Description

This is an internal function that processes a list of test files for 'longtests'. It loops over each test file and extracts the 'longtests' and 'functionname' fields. It then prepares the tests, optionally indents the code, and formats the tests into a string with an optional 'test_that' boilerplate. The function also constructs a header for the content, including a 'context' line if specified. The resulting content strings are returned in a named list, with the names being the paths of the test files.

Usage

```
process_testfiles(
  testfiles,
  indent_code = FALSE,
  add_testthat_boilerplate = FALSE,
  add_context_header = FALSE
)
```

Arguments

`testfiles`	A named list of test files, where the names are the paths of the files.
`indent_code`	A logical value indicating whether to indent the code in the tests.
`add_testthat_boilerplate`	A logical value indicating whether to add a 'test_that' boilerplate to the tests.
`add_context_header`	A logical value indicating whether to add a 'context' line to the header.

Value

A named list of content strings for the test files.

`roclet_clean.roclet_longtests`
Clean up test files in the 'longtests' directory

Description

This function cleans up test files in the 'longtests' directory of a given base path. It first verifies if 'longtests' is used in the package by calling `verify_longtests_used()`. Then, it finds all test files in the 'longtests' directory that match the pattern "test-biocroxytest-.*\.R$". Finally, it calls `internal_longtests_roclet_clean(testfiles)` to remove the test files generated by biocroxytest.

Usage

```
## S3 method for class 'roclet_longtests'
roclet_clean(x, base_path)
```

Arguments

x	An object of class 'roclet_longtests'.
base_path	A character string representing the base path of the package.

Value

Invisible NULL.

Examples

```
# Set up a temporary directory
base_path <- tempdir()
longtests_path <- file.path(base_path, "longtests")
unlink(longtests_path, recursive = TRUE, force = TRUE)
dir.create(longtests_path, recursive = TRUE, showWarnings = FALSE)

# Create dummy inputs
obj <- longtests_roclet()

# Run the roclet_output function
result <- roxygen2::roclet_clean(obj, base_path)
```

---

roclet_output.roclet_longtests

Process blocks for 'longtests' roclet

Description

This function processes a list of blocks for the 'longtests' roclet. It calls the 'internal_longtests_roclet_process' function with the specified blocks and additional arguments. The code in the tests is indented, a 'test_that' boilerplate is added to the tests, and a 'context' line is not added to the header.

Usage

```
## S3 method for class 'roclet_longtests'
roclet_output(x, results, base_path, ...)
```

Arguments

x	An object of class 'roclet_longtests'. package.
results	A list with a single element 'longtests', which contains the
base_path	A character string representing the base path of the
...	Additional arguments passed to the function.

Value

A list with a single element 'longtests', which contains the processed test files.

Examples

```
# Set up a temporary directory
base_path <- tempdir()
longtests_path <- file.path(base_path, "longtests", "testthat")
unlink(longtests_path, recursive = TRUE, force = TRUE)
dir.create(longtests_path, recursive = TRUE, showWarnings = FALSE)

# Create dummy inputs
obj <- longtests_roclet()
results <- list(longtests = list())

# Run the roclet_output function
result <- roxygen2::roclet_output(obj, results, base_path)
```

---

roclet_process.roclet_longtests

Process blocks for 'longtests' roclet

Description

This function processes a list of blocks for the 'longtests' roclet. It calls the 'internal_longtests_roclet_process' function with the specified blocks and additional arguments. The code in the tests is indented, a 'test_that' boilerplate is added to the tests, and a 'context' line is not added to the header.

Usage

```
## S3 method for class 'roclet_longtests'
roclet_process(x, blocks, env, base_path)
```

Arguments

x	An object of class 'roclet_longtests'.
blocks	A list of roxygen2 blocks.
env	The environment in which the roxygen2 blocks were parsed.
base_path	A character string representing the base path of the package.

Value

A list with a single element 'longtests', which contains the processed test files.

Examples

```
# Create a dummy block function
block <- function(roclet, value) {
  list(roclet = roclet, value = value)
}

# Create a new roclet_longtests object
x <- longtests_roclet()

# Define some roxygen2 blocks
blocks <- list(
  block(
    roclet = "longtests",
    value = list(
      raw = "test_that('this is a test', { expect_equal(1, 1) })"
    )
  )
)

blocks <- structure(blocks, class = "roclet_longtests")

# Define the environment and base path
env <- globalenv()
base_path <- tempdir()

# Process the blocks
roxygen2::roclet_process(x, blocks, env, base_path)
```

`roxy_tag_parse.roxy_tag_longtests`
Parse a roxy_tag_longtests object

Description

This function parses a 'roxy_tag_longtests' object. It first checks if the object inherits from 'roxy_tag_longtests' and 'roxy_tag'. If not, it aborts with an error message. Then, it checks if the 'raw' field of the object is empty. If it is, it returns a warning that a value is required. Finally, it removes any leading newline characters from the 'raw' field and assigns the result to the 'val' field of the object.

Usage

```
## S3 method for class 'roxy_tag_longtests'
roxy_tag_parse(x)
```

Arguments

x	An object of class 'roxy_tag_longtests'.

Value

The input object with the 'val' field updated.

`use_longtests` *Sets up overall longtests infrastructure*

Description

This function is used to set up the environment for running long tests. It calls two helper functions: `setup_bboptions()` and `setup_longtetsts()`.

Usage

```
use_longtests()
```

Details

The `use_longtests()` function is a wrapper function that calls `setup_bboptions()` and `setup_longtetsts()`. The `setup_bboptions()` function checks if the .BBSoptions file exists and creates it if it doesn't. It then checks the contents of the .BBSoptions file and adds or modifies the 'RunLongTests: TRUE' line as needed. The `setup_longtetsts()` function creates a 'longtests/testthat' directory if it doesn't exist and copies the 'tests/testthat.R' file into it. If the 'tests/testthat.R' file doesn't exist, it creates a new one with default content.

Value

This function does not return a value. It is used for its side effects of setting up the environment for running long tests.

Examples

```
# Create the longtests directory and .BBSoptions file
use_longtests()

# Remove the longtests directory and .BBSoptions file
unlink(file.path(".BBSoptions"), recursive = TRUE, force = TRUE)
unlink(file.path("longtests"), recursive = TRUE, force = TRUE)
```

`verify_longtests_used` *Check if longtests are used in the package*

Description

This function checks if the directory "longtests" exists in the current package. If it does not exist, the function will throw an error message and suggest setting it up using the `biocroxytest::use_longtests()` function.

Usage

```
verify_longtests_used(base_path = ".")
```

Arguments

`base_path`	The base path of the package.

Value

TRUE if "longtests" directory exists, FALSE otherwise.

Index