

# Package: biocroxytest (via r-universe)

June 30, 2024

**Title** Handle Long Tests in Bioconductor Packages

**Version** 1.1.0

**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 (>= 4.3.0)

**Config/testthat/edition** 3

**Config/testthat/parallel** false

**VignetteBuilder** knitr

**Repository** <https://bioc.r-universe.dev>

**RemoteUrl** <https://github.com/bioc/biocroxytest>

**RemoteRef** HEAD

**RemoteSha** 59938137a1cdc3bffbfc3e74c38e55ea8c34171e

## Contents

longtests_roclet . . . . .	2
roclet_clean.roclet_longtests . . . . .	3
roclet_output.roclet_longtests . . . . .	4
roclet_process.roclet_longtests . . . . .	5
use_longtests . . . . .	6
<b>Index</b>	<b>7</b>

---

longtests_roclet	<i>Roclet for Long Tests</i>
------------------	------------------------------

---

## 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()
```

---

`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

<code>x</code>	An object of class 'roclet_longtests'. package.
<code>results</code>	A list with a single element 'longtests', which contains the
<code>base_path</code>	A character string representing the base path of the
<code>...</code>	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::rocllet_process(x, blocks, env, base_path)
```

---

use_longtests	<i>Sets up overall longtests infrastructure</i>
---------------	---

---

### 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)
```

# Index

`longtests_roclet`, [2](#)

`roclet_clean.roclet_longtests`, [3](#)

`roclet_output.roclet_longtests`, [4](#)

`roclet_process.roclet_longtests`, [5](#)

`use_longtests`, [6](#)