Package ‘testex’

April 15, 2024

Title  Add Tests to Examples
Version  0.2.0
Description  Add tests in-line in examples. Provides standalone functions for facilitating easier test writing in Rd files. However, a more familiar interface is provided using 'roxygen2' tags. Tools are also provided for facilitating package configuration and use with 'testthat'.

License  MIT + file LICENSE
Depends  R (>= 3.2.0)
Imports  utils
Suggests  testthat, withr, callr, roxygen2, spelling, knitr, rmarkdown
Encoding  UTF-8
RoxygenNote  7.3.1
Language  en-US
VignetteBuilder  knitr
Config/testex/options  list(version = "0.2.0")
NeedsCompilation  no
Author  Doug Kelkhoff [aut, cre]
Maintainer  Doug Kelkhoff <doug.kelkhoff@gmail.com>
Repository  CRAN
Date/Publication  2024-04-14 23:30:02 UTC

R topics documented:

fallback_expect_no_error ............................................. 2
testex ................................................................. 3
testex-roxygen-tags .................................................. 4
testex-testthat ....................................................... 5
test_examples_as_testthat ........................................... 6
uses_roxygen2 .......................................................... 7
**fallback_expect_no_error**

*Expect no Error*

**Description**

Expect no Error

**Usage**

```r
fallback_expect_no_error(object, ...)
```

**Arguments**

- `object`  
  An expression to evaluate
- `...`  
  Additional arguments unused

**Value**

The value produced by the expectation code

**Note**

This is a stop-gap implementation, and will only be used for legacy versions of testthat before this was properly supported.

A testthat expectation that the provided code can be evaluated without producing an error. This is the most basic expectation one should expect of any example code. Further expectations are provided in subsequent testthat code.
testex  
A syntactic helper for writing quick and easy example tests

Description

A wrapper around stopifnot that allows you to use . to refer to .Last.value and preserve the last non-test output from an example.

Usage

testex(
  ..., 
  srcref = NULL, 
  example_srcref = NULL, 
  value = get_example_value(), 
  envir = parent.frame(), 
  style = "standalone"
)

Arguments

... Expressions to evaluated. . will be replaced with the expression passed to val, and may be used as a shorthand for the last example result.

srcref An option srcref_key string used to indicate where the relevant test code originated from.

eexample_srcref An option srcref_key string used to indicate where the relevant example code originated from.

value A value to test against. By default, this will use the example’s .Last.value.

eenvir An environment in which tests should be evaluated. By default the parent environment where tests are evaluated.

style A syntactic style used by the test. Defaults to "standalone", which expects TRUE and uses a -.notation. Accepts one of "standalone" or "testthat". By default, styles will be implicitly converted to accommodate known testing frameworks, though this can be disabled by passing the style "AsIs" with I().

Value

Invisibly returns the .Last.value as it existed prior to evaluating the test.

Documenting with testex

testex is a simple wrapper around execution that propagates the .Last.value returned before running, allowing you to chain tests more easily.

Use in Rd files::


\examples{
  f <- function(a, b) a + b
  f(3, 4)
  \testonly{
    testex::testex(
      is.numeric(.),
      identical(. , 7)
    )
  }
}

But Rd files are generally regarded as being a bit cumbersome to author directly. Instead, testex provide helpers that generate this style of documentation, which use this function internally.

Use with roxygen2:
Within a roxygen2 @examples block you can instead use the @test tag which will generate Rd code as shown above.

# @examples
#' f <- function(a, b) a + b
#' f(3, 4)
#' @test is.numeric(.)
#' @test identical(. , 7)

---

testex provides two new roxygen2 tags, @test and @testthat.

tags
testex tags are all sub-tags meant to be used within an @examples block. They should be considered as tags within the @examples block and used to construct blocks of testing code within example code.

@test:  In-line expectations to test the output of the previous command within an example. If . is used within the test expression, it will be used to refer to the output of the previous example command. Otherwise, the result of the expression is expected to be identical to the previous output.

# @examples
#' 1 + 2
#' @test 3
#' @test . == 3
#'
# @examples
#' 3 + 4
#' @test identical(. , 7)
@testthat: Similar to @test, @testthat can be used to make in-line assertions using testthat expectations. testthat expectations follow a convention where the first argument is an object to compare against an expected value or characteristic. Since the value will always be the result of the previous example, this part of the code is implicitly constructed for you.

If you want to use the example result elsewhere in your expectation, you can refer to it with a `. `. When used in this way, testex will not do any further implicit modification of your expectation.

```r
# @examples
# `1 + 2
# @testthat expect_equal(3)
# @testthat expect_gt(0)
#
# @examples
# `3 + 4
# @testthat expect_equal(., 7)
```

---

testex-testthat Support for testthat Expectations

### Description

testthat support is managed through a "style" provided to testex. When using the testthat style (automatically when using the @testthat tag), expectations are processed such that they always refer to the previous example. Special care is taken to manage propagation of this value through your test code, regardless of how testthat is executed.

### Examples

```r
# example code
1 + 2

# within `testex` block, test code refers to previous result with `.`
testex(style = "testthat", srcref = "abc.R:1:3", {
  test_that("addition holds up", {
    expect_equal(., 3)
  })
})
```
test_examples_as_testthat

Execute examples from Rd files as testthat tests

Description

Reads examples from Rd files and constructs testthat-style tests. testthat expectations are built such that

Usage

test_examples_as_testthat(
  package,
  path,
  ...,  
  test_dir = file.path(tempdir(), "testex-tests"),
  clean = TRUE,
  overwrite = TRUE,
  roxygenize = !is_r_cmd_check() && uses_roxygen2(path),
  reporter = testthat::get_reporter()
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>A package name whose examples should be tested</td>
</tr>
<tr>
<td>path</td>
<td>Optionally, a path to a source code directory to use. Will only have an effect if parameter package is missing.</td>
</tr>
<tr>
<td>...</td>
<td>Additional argument unused</td>
</tr>
<tr>
<td>test_dir</td>
<td>An option directory where test files should be written. Defaults to a temporary directory.</td>
</tr>
<tr>
<td>clean</td>
<td>Whether the test_dir should be removed upon completion of test execution. Defaults to TRUE.</td>
</tr>
<tr>
<td>overwrite</td>
<td>Whether files should be overwritten if test_dir already exists. Defaults to TRUE.</td>
</tr>
<tr>
<td>roxygenize</td>
<td>Whether R documentation files should be re-written using roxygen2 prior to testing. When not FALSE, tests written in roxygen2 tags will be</td>
</tr>
<tr>
<td></td>
<td>used to update R documentation files prior to testing to use the most up-to-date example tests. May be TRUE, or a list of arguments passed</td>
</tr>
<tr>
<td></td>
<td>to roxygen2::roxygenize. By default, only enabled when running outside of R CMD check and while taking roxygen2 as a dependency.</td>
</tr>
<tr>
<td>reporter</td>
<td>A testthat reporter to use. Defaults to the active reporter in the testthat environment or default reporter.</td>
</tr>
</tbody>
</table>
uses_roxygen2

Details

1. Each example expression is expected to run without error
2. Any testex expectations are expected to pass

Generally, you won’t need to use this function directly. Instead, it is called by a file generated by `use_testex_as_testthat()` which will add any testex example tests to your existing testthat testing suite.

Value

The result of `testthat::source_file()`, after iterating over generated test files.

Note

It is assumed that this function is used within a testthat run, when the necessary packages are already installed and loaded.

Examples

```r
# library(pkg.example)
path <- system.file("pkg.example", package = "testex")
test_examples_as_testthat(path = path)
```

---

**uses_roxygen2**  
*Checks for use of roxygen2*

Description

Checks for use of roxygen2

Usage

`uses_roxygen2(path)`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
<td>A file path to a package source code directory</td>
</tr>
</tbody>
</table>

Value

A logical value indicating whether a package takes roxygen2 as a dependency.
use_testex

Add testex tags and configure package to fully use testex features

Description
Add testex tags and configure package to fully use testex features

Usage
use_testex(path = getwd(), check = TRUE, quiet = FALSE)

Arguments
- path: A package source code working directory
- check: A logical value indicating whether tests should be executing during R CMD check.
- quiet: Whether output should be suppressed

Value
The result of write.dcf() upon modifying the package DESCRIPTION file.

Note
The testex roxygen2 tags behave similarly to roxygen2 @examples tags, with the minor addition of some wrapping code to manage the tests. This means that they will be integrated into your @examples and can be intermixed between @examples tags

use_testex_as_testthat

Run examples as testthat expectations

Description
Run examples as testthat expectations

Usage
use_testex_as_testthat(path = getwd(), context = "testex", quiet = FALSE)

Arguments
- path: A package source code working directory
- context: A testthat test context to use as the basis for a new test filename.
- quiet: Whether to emit output messages.
with_attached

Value
The result of writeLines() after writing a new testthat file.

with_attached  Temporarily attach a namespace

Description
This function is primarily for managing attaching of namespaces needed for testing internally. It is exported only because it is needed in code generated within Rd files, but is otherwise unlikely to be needed.

Usage
with_attached(ns, expr)

Arguments
ns  A namespace or namespace name to attach
expr  An expression to evaluate while the namespace is attached

Value
The result of evaluating expr

with_srcref  Raise testthat Expectations With A Known Source Reference

Description
Retroactively assigns a source file and location to a expectation. This allows testthat to report an origin for any code that raised an example test failure from the source roxygen2 code, even though the test code is reconstructed from package documentation files.

Usage
with_srcref(src, expr, envir = parent.frame())

Arguments
src  A srcref_key which is parsed to produce an artificial srcref for the expectation signaled messages.
expr  An expression to be evaluated. If an expectation condition is raised during its evaluation, its srcref is converted to src.
envir  An environment in which to evaluate expr.
Value

The result of evaluating `expr`, or an expectation with appended `srcref` information if an expectation is raised.
Index

* use
  use_testex_as_testthat, 8

fallback_expect_no_error, 2

I(), 3

roxygen2::roxygenize, 6

srcref, 9, 10

test_examples_as_testthat, 6
testex, 3, 4, 5, 8
testex-roxygen-tags, 4
testex-testthat, 5
testthat::source_file(), 7

use_testex, 8
use_testex_as_testthat, 8
use_testex_as_testthat(), 7
uses_roxygen2, 7

with_attached, 9
with_srcref, 9
write.dcf(), 8
writeLines(), 9