

Package: autotest (via r-universe)

September 13, 2026

Title Automatic Package Testing

Version 0.2.0.003

Description Automatic testing of R packages by tracing function calls made in examples and tests, then systematically varying the inputs to check that functions respond sensibly. The package is a development aid for authors submitting to the 'rOpenSci' statistical software review project ('rOpenSci' authors, 2021; <[doi:10.5281/zenodo.5556756](https://doi.org/10.5281/zenodo.5556756)>).

License GPL-3

URL <https://docs.ropensci.org/autotest/>,
<https://github.com/ropensci-review-tools/autotest>

BugReports <https://github.com/ropensci-review-tools/autotest/issues>

Imports checkmate, cli, data.table, fs, grDevices, here, magrittr, memoise, methods, pkgload, rlang, testthat, tibble, typetracer, withr

Suggests devtools, dplyr, DT, goodpractice, knitr, rmarkdown, roxygen2, tidyr, usethis

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

Config/ropensci/maintainer staff

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://ropensci.r-universe.dev>

Date/Publication 2026-09-13 13:37:17 UTC

RemoteUrl <https://github.com/ropensci-review-tools/autotest>

RemoteRef main

RemoteSha eb1f646d58a5e1e4f0ee4452e9d506fb34529cd6

Contents

autotest_obj	2
autotest_package	3
autotest_types	5
expect_autotest_no_err	6
expect_autotest_no_testdata	6
expect_autotest_no_warn	7
expect_autotest_notes	8
expect_autotest_testdata	8
Index	10

autotest_obj	<i>autotest_obj class definition</i>
--------------	--------------------------------------

Description

This function exists only to provide the class definitions for test objects, and is not intended to be called directly.

Usage

```
autotest_obj(  
  package = NA_character_,  
  package_loc = NULL,  
  test_name = NA_character_,  
  fn_name = NA_character_,  
  parameters = list(),  
  parameter_types = NA_character_,  
  class = NULL,  
  classes = NULL,  
  env = new.env(),  
  test = FALSE,  
  quiet = FALSE  
)
```

Arguments

package	Name of package for which object is to be constructed.
package_loc	Location of package on local system (for source packages only)
test_name	Name of test (use autotest_types to get all test names).
fn_name	Name of function to be tested.
parameters	Names of all parameters for that function.
parameter_types	Types of input parameters.
class	Class of an individual parameter.

classes	Classes of all parameters.
env	Environment in which tests are to be run.
test	If FALSE, return only descriptions of tests which would be run with test = TRUE, without actually running them.
quiet	If FALSE, issue progress and other messages during testing of object.

Value

An object of class `autotest_obj`, a list holding the parameters passed as arguments, used internally to represent one autotest test case.

Examples

```
x <- autotest_obj (
  package = "stats",
  fn_name = "sd",
  test_name = "test"
)
```

autotest_package	<i>autotest_package</i>
------------------	-------------------------

Description

Automatically test an entire package by tracing calls made in its documented examples (and, for local source packages, its test suite) with **typetracer**, then testing each traced function call in turn.

Usage

```
autotest_package(
  package = ".",
  functions = NULL,
  exclude = NULL,
  test = FALSE,
  test_data = NULL,
  progress = c("bar", "tests", "none")
)
```

Arguments

package	Name of package, as either 1. Path to local package source 2. Name of installed package 3. Full path to location of installed package if not on .libPaths, or 4. Default which presumes current directory is within package to be tested.
---------	--

functions	Optional character vector containing names of functions of nominated package to be included in 'autotesting'.
exclude	Optional character vector containing names of any functions of nominated package to be excluded from 'autotesting'.
test	If FALSE, return only descriptions of tests which would be run with test = TRUE, without actually running them.
test_data	Result returned from calling either autotest_types or autotest_package with test = FALSE that contains a list of all tests which would be conducted. These tests have an additional flag, test, which defaults to TRUE. Setting any tests to FALSE will avoid running them when test = TRUE.
progress	Style of progress display while testing functions, one of: • "bar" (default) A cli progress bar. Automatically falls back to "none" when called from within a knitr document, to avoid literal ANSI escape sequences leaking into the rendered output. • "tests" One line per function tested, showing [i / n]. • "none" No progress display at all.

Value

An `autotest_package` object which is derived from a **tibble** `tbl_df` object. This has one row for each test, and the following eight columns:

1. type The type of result, either "dummy" for test = FALSE, or one of "error", "warning", "diagnostic", or "message".
2. test_name Name of each test
3. fn_name Name of function being tested
4. parameter Name of parameter being tested
5. parameter_type Expected type of parameter as identified by autotest.
6. operation Description of the test
7. content For test = FALSE, the expected behaviour of the test; for test = TRUE, the observed discrepancy with that expected behaviour
8. test If FALSE (default), list all tests without implementing them, otherwise implement all tests.

Some columns may contain NA values, as explained in the Note.

Note

Some columns may contain NA values, including:

- `parameter` and `parameter_type`, for tests applied to entire functions, such as tests of return values.
- `test_name` for warnings or errors generated through "normal" function calls generated directly from example code, in which case type will be "warning" or "error", and content will contain the content of the corresponding message.

See Also

Other main_functions: [autotest_types\(\)](#)

Examples

```
x <- autotest_package (package = "stats", functions = "var", test = FALSE)
x
```

autotest_types	<i>autotest_types</i>
----------------	-----------------------

Description

List all types of 'autotests' currently implemented.

Usage

```
autotest_types(notest = NULL)
```

Arguments

notest	Character string of names of tests which should be switched off by setting the test column to FALSE. Run this function first without this parameter to get all names, then re-run with this parameter switch specified tests off.
--------	---

Value

An autotest object with each row listing one unique type of test which can be applied to every parameter (of the appropriate class) of each function.

See Also

Other main_functions: [autotest_package\(\)](#)

Examples

```
x <- autotest_types ()
x
```

```
expect_autotest_no_err
      expect_autotest_no_err
```

Description

Expect `autotest_package()` to be clear of errors

Usage

```
expect_autotest_no_err(object)
```

Arguments

`object` An autotest object to be tested

Value

(invisibly) The same object

See Also

Other expectations: [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

Examples

```
x <- autotest_package (package = "stats", functions = "cov", test = TRUE)
testthat::expect_failure (expect_autotest_no_err (x))
```

```
expect_autotest_no_testdata
      expect_autotest_no_testdata
```

Description

Expect `autotest_package()` to be clear of errors with no tests switched off

Usage

```
expect_autotest_no_testdata(object = NULL)
```

Arguments

`object` Not used here, but required for `testthat` expectations

Value

(invisibly) The autotest object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

Examples

```
## Not run:  
# Called within a 'testthat' test file of the local package itself:  
testthat::test_that ("autotest", {  
  testthat::expect_success (expect_autotest_no_testdata ())  
})  
  
## End(Not run)
```

expect_autotest_no_warn

expect_autotest_no_warn

Description

Expect `autotest_package()` to be clear of warnings

Usage

```
expect_autotest_no_warn(object)
```

Arguments

object An autotest object to be tested

Value

(invisibly) The same object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

Examples

```
x <- autotest_package (package = "stats", functions = "cov", test = TRUE)  
testthat::expect_failure (expect_autotest_no_warn (x))
```

expect_autotest_notes *expect_autotest_notes*

Description

Expect test_data param of autotest_package to have additional note column explaining why tests have been switched off.

Usage

```
expect_autotest_notes(object)
```

Arguments

object An autotest object to be tested

Value

(invisibly) The same object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_testdata\(\)](#)

Examples

```
x <- autotest_package (package = "stats", functions = "cov", test = TRUE)
testthat::expect_success (expect_autotest_notes (x))
```

expect_autotest_testdata
 expect_autotest_testdata

Description

Expect autotest_package() to be clear of errors with some tests switched off, and to have note column explaining why those tests are not run.

Usage

```
expect_autotest_testdata(object)
```

Arguments

object An autotest_package object with a test column flagging tests which are not to be run on the local package.

Value

(invisibly) The autotest object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#)

Examples

```
test_data <- autotest_types (notest = "vector_to_list_col")
## Not run:
# Called within a 'testthat' test file of the local package itself:
testthat::test_that ("autotest", {
  testthat::expect_success (expect_autotest_testdata (test_data))
})

## End(Not run)
```

Index

* **class**

autotest_obj, [2](#)

* **expectations**

expect_autotest_no_err, [6](#)

expect_autotest_no_testdata, [6](#)

expect_autotest_no_warn, [7](#)

expect_autotest_notes, [8](#)

expect_autotest_testdata, [8](#)

* **main_functions**

autotest_package, [3](#)

autotest_types, [5](#)

.libPaths, [3](#)

autotest_obj, [2](#)

autotest_package, [3](#), [4](#)

autotest_package(), [5](#)

autotest_types, [2](#), [4](#), [5](#)

autotest_types(), [5](#)

expect_autotest_no_err, [6](#)

expect_autotest_no_err(), [7–9](#)

expect_autotest_no_testdata, [6](#)

expect_autotest_no_testdata(), [6–9](#)

expect_autotest_no_warn, [7](#)

expect_autotest_no_warn(), [6–9](#)

expect_autotest_notes, [8](#)

expect_autotest_notes(), [6](#), [7](#), [9](#)

expect_autotest_testdata, [8](#)

expect_autotest_testdata(), [6–8](#)