

# Snapshot tests



Drawing from newly updated  
<https://r-pkgs.org/tests.html>

*July 2022*



Jenny Bryan  
@jennybryan twitter, @jennybc github  
Software Engineer, RStudio



testthat setup: once per package\*

```
usethis::use_testthat(3)
```

\*ok, technically, once per package, per testthat edition

# testthat 3e

Snapshot tests are arguably the flagship new feature.

The use of the new-ish waldo package for comparing "this vs. that" ("actual vs. expected", usually) is also pretty nifty.

I shall not today attempt further to define  
"hard-core pornography", and perhaps I  
could never succeed in intelligibly doing so.

But I know it when I see it, and the motion  
picture involved in this case is not that.

US Supreme Court Justice Potter Stewart

I shall not today attempt further to define  
**this test's expected result**, and perhaps I  
could never succeed in intelligibly doing so.

But I know it when I see it, and **the actual  
result** we're getting today is not that.

Your failing snapshot test

Big idea:

Expected result is captured once and stored as a file.

Future test runs compare actual result to that file snapshot.

Especially suitable for, e.g., testing messages and errors.

# Example: how waldo reports differences

```
withr::with_options(  
  list(width = 20),  
  waldo::compare(c("X", letters), c(letters, "X"))  
)  
#>      old | new  
#> [1] "X" -  
#> [2] "a" | "a" [1]  
#> [3] "b" | "b" [2]  
#> [4] "c" | "c" [3]  
#>  
#>      old | new  
#> [25] "x" | "x" [24]  
#> [26] "y" | "y" [25]  
#> [27] "z" | "z" [26]  
#>      - "X" [27]
```

# How to apply a snapshot test the example

```
test_that("side-by-side diffs work", {  
  withr::local_options(width = 20)  
  expect_snapshot(  
    waldo::compare(c("X", letters), c(letters, "X"))  
  )  
})
```



# Interactive execution of snapshot tests doesn't "work"

Can't compare snapshot to reference when testing interactively

i Run ``devtools::test()`` or ``testthat::test_file()`` to see changes

It is harmless to execute snapshot tests interactively.

But it's a no-op.

No snapshot recording or comparison happens.

# Snapshot tests only work in a non-interactive context

Snapshot test recording and comparison only happen when the tests are being run at arms-length, via some sort of automated process.

For example: running an entire test file, running the entire test suite.

# New snapshot! Warning is normal

— Warning (test-diff.R:63:3): side-by-side diffs work

---

Adding new snapshot:

Code

```
waldo::compare(c(
  "X", letters), c(
  letters, "X"))
```

Output

	old		new
--	-----	--	-----

[1]	"X"	-	
-----	-----	---	--

[2]	"a"		"a" [1]
-----	-----	--	---------

[3]	"b"		"b" [2]
-----	-----	--	---------

[4]	"c"		"c" [3]
-----	-----	--	---------

	old		new
--	-----	--	-----

[25]	"x"		"x" [24]
------	-----	--	----------

[26]	"y"		"y" [25]
------	-----	--	----------

[27]	"z"		"z" [26]
------	-----	--	----------

	-	"X"	[27]
--	---	-----	------

# When snapshot tests fail

— Failure (test-diff.R:63:3): side-by-side diffs work

---

Snapshot of code has changed:

old[3:15] vs new[3:15]

```
"      \"X\", letters), c("
"      letters, \"X\"))"
"Output"
```

```
- "      old | new      "
+ "      OLD | NEW      "
"  [1] \"X\" -          "
"  [2] \"a\" | \"a\" [1]"
"  [3] \"b\" | \"b\" [2]"
"  [4] \"c\" | \"c\" [3]"
"  "
```

```
- "      old | new      "
+ "      OLD | NEW      "
```

and 3 more ...

\* Run ``snapshot_accept('diff')`` to accept the change

\* Run ``snapshot_review('diff')`` to interactively review the change

# Reacting to snapshot test failure (change, really)

- \* Run ``snapshot_accept('diff')`` to accept the change
- \* Run ``snapshot_review('diff')`` to interactively review the change

# Key arguments to `expect_snapshot()`

1. `cran = FALSE`

2. `error = FALSE`

3. `transform`

4. `variant`

# Various ways to use snapshot tests with errors

- If you want to inspect the error message, use `expect_snapshot(error = TRUE)`.
- If you want to check the class, use `expect_error(class = )`.
- If you want to expect the message and check the class, use `expect_snapshot_error(class = )`.

<https://github.com/r-lib/testthat/issues/1471#issuecomment-1005372257>

<https://github.com/r-lib/testthat/issues/1548>

<https://github.com/r-lib/testthat/pull/1616>