

## report-time

The tracking issue for this feature is: #64888

---

The `report-time` feature adds a possibility to report execution time of the tests generated via `libtest`.

This is unstable feature, so you have to provide `-Zunstable-options` to get this feature working.

Sample usage command:

```
./test_executable -Zunstable-options --report-time
```

Available options:

```
--report-time    Show execution time of each test.
                  Threshold values for colorized output can be
                  configured via
                  `RUST_TEST_TIME_UNIT`, `RUST_TEST_TIME_INTEGRATION`
                  and
                  `RUST_TEST_TIME_DOCTEST` environment variables.
                  Expected format of environment variable is
                  `VARIABLE=WARN_TIME,CRITICAL_TIME`.
                  Not available for --format=terse

--ensure-time     Treat excess of the test execution time limit as
                  error.
                  Threshold values for this option can be configured via
                  `RUST_TEST_TIME_UNIT`, `RUST_TEST_TIME_INTEGRATION`
                  and
                  `RUST_TEST_TIME_DOCTEST` environment variables.
                  Expected format of environment variable is
                  `VARIABLE=WARN_TIME,CRITICAL_TIME`.
                  `CRITICAL_TIME` here means the limit that should not be
                  exceeded by test.
```

Example of the environment variable format:

```
RUST_TEST_TIME_UNIT=100,200
```

where 100 stands for warn time, and 200 stands for critical time.

## Examples

```
cargo test --tests -- -Zunstable-options --report-time
Finished dev [unoptimized + debuginfo] target(s) in 0.02s
Running target/debug/deps/example-27fb188025bec02c
```

```
running 3 tests
test tests::unit_test_quick ... ok <0.000s>
test tests::unit_test_warn ... ok <0.055s>
test tests::unit_test_critical ... ok <0.110s>
```

```
test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out
```

```
Running target/debug/deps/tests-cedb06f6526d15d9
```

```
running 3 tests
test unit_test_quick ... ok <0.000s>
test unit_test_warn ... ok <0.550s>
test unit_test_critical ... ok <1.100s>
```

```
test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out
```