

Teuthida Technologies Home

All about embedded systems engineering.

Often, Less is Better!

Posted on [April 1, 2020](#)

When I was learning Ruby, I was searching for things that could be improved. I was looking for ways I could make things better. One of the areas I was involved in was testing. This should come as no surprise. If you are writing Ruby code, you need to be writing tests. The exact order in which this is done is not my point. Just that tests must be written and run frequently. Here is a typical test run using the *minitest* facility:

```
23 mysh>ut
Run options: --seed 64999

# Running:

.....

Finished in 0.014689s, 1429.6325 runs/s, 6263.1518 assertions/s.
21 runs, 92 assertions, 0 failures, 0 errors, 0 skips
C:\Sites\full_dup
24 mysh>
```

— Standard Test Output

To my mind, this was not communicating that true progress of the tests. Consider: What version of *minitest* was in use? What files were being processed? How many tests were in each of those files? Most of all, why does this output look so boring?

Yes; boring! So being bored I thought: Surely I can do better! I came up with a gem I call *minitest_visible* to spruce things up. Here's what its output looks like:

```
C:\Sites\full_dup
20 mysh>ut
Run options: --seed 15356

# Running:

MiniTest version = 5.11.3

Running test file: array_tests.rb
.....
Running test file: struct_tests.rb
.....
Running test file: hash_tests.rb
.....
Running test file: object_tests.rb
.....
Running test file: full_dup_tests.rb
.

Finished in 0.027631s, 760.0168 runs/s, 3329.5973 assertions/s.
21 runs, 92 assertions, 0 failures, 0 errors, 0 skips
C:\Sites\full_dup
21 mysh>
```

— Enhanced Test Output

Much better! Clearer! Nice progress! and so entertaining? Well at least I thought it was a big improvement. It would seem that not many people shared my view. I can see their points. Notice that the bottom line is the same in both cases:

21 runs, 92 assertions, 0 failures, 0 errors, 0 skips

This result line is what counts. These are tests, not video games. They should test the code without a lot of bells and whistles and let you get on with the real work.

I was also told that my little gem did not fit in with the *minitest* ecosystem. I asked what was meant by this but I never got a reply. I suppose somebody was channeling their inner Linus Torvalds of rudeness that day. I'll never know. It also does not matter.

I have finally come to realize that the critics were right. The best answer is to keep things as simple and lean as possible. No being fancy; no showing off. To that end I have worked to remove the *minitest_visible* augmentation from almost all of my work.

Affected gems are: *composite_rng*, *counted_cache*, *fibonacci_rng*, *flex_array*, *format_engine*, *full_clone*, *full_dup*, *fully_freeze*, *in_array*, *insouciant*, *lexical_analyzer*, *make_gem*, *mini_erb*, *mini_readline*, *mini_term*, *mysh*, *parse_queue*, *pause_output*, *safe_clone*, and *safe_dup*. The exception is the *fOOrth* gem experiment that has so many test files that it would be hard to sort out what was happening without a little help. At least for now.

Now these gems can be used by others without the hassle of having to install a non-standard testing extension.

Is less more? I'm not sure. Is less better? Very often, yes it is.

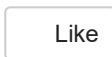
Your truly

Peter Camilleri (aka Squidly Jones)

SHARE THIS:



LIKE THIS:



Be the first to like this.

This entry was posted in [Announcements](#), [Hidden Traps](#), [Ruby](#) by [Peter Camilleri](#). Bookmark the [permalink \[http://teuthida-technologies.com/?p=1881\]](http://teuthida-technologies.com/?p=1881) .

