



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'Test2::Formatter.3perl'***

***\$ man Test2::Formatter.3perl***

Test2::Formatter(3perl) Perl Programmers Reference Guide Test2::Formatter(3perl)

**NAME**

Test2::Formatter - Namespace for formatters.

**DESCRIPTION**

This is the namespace for formatters. This is an empty package.

**CREATING FORMATTERS**

A formatter is any package or object with a "write(\$event, \$num)" method.

```
package Test2::Formatter::Foo;

use strict;

use warnings;

sub write {

    my $self_or_class = shift;

    my ($event, $assert_num) = @_;

    ...

}

sub hide_buffered { 1 }

sub terminate { }

sub finalize { }

sub supports_tables { return $BOOL }

sub new_root {

    my $class = shift;

    ...

    $class->new(@_);
```

```
}  
1;
```

The "write" method is a method, so it either gets a class or instance. The two arguments are the \$event object it should record, and the \$assert\_num which is the number of the current assertion (ok), or the last assertion if this event is not itself an assertion.

The assertion number may be any integer 0 or greater, and may be undefined in some cases.

The "hide\_buffered()" method must return a boolean. This is used to tell buffered subtests whether or not to send it events as they are being buffered. See "run\_subtest(...)" in Test2::API for more information.

The "terminate" and "finalize" methods are optional methods called that you can implement if the format you're generating needs to handle these cases, for example if you are generating XML and need close open tags.

The "terminate" method is called when an event's "terminate" method returns true, for example when a Test2::Event::Plan has a 'skip\_all' plan, or when a Test2::Event::Bail event is sent. The "terminate" method is passed a single argument, the Test2::Event object which triggered the terminate.

The "finalize" method is always the last thing called on the formatter, except when "terminate" is called for a Bail event. It is passed the following arguments:

The "supports\_tables" method should be true if the formatter supports directly rendering table data from the "info" facets. This is a newer feature and many older formatters may not support it. When not supported the formatter falls back to rendering "detail" instead of the "table" data.

The "new\_root" method is used when constructing a root formatter. The default is to just delegate to the regular "new()" method, most formatters can ignore this.

- ? The number of tests that were planned
- ? The number of tests actually seen
- ? The number of tests which failed
- ? A boolean indicating whether or not the test suite passed
- ? A boolean indicating whether or not this call is for a subtest

The "new\_root" method is called when "Test2::API::Stack" initializes the root hub for the first time. Most formatters will simply have this call "\$class->new", which is the default behavior. Some formatters however may want to take extra action during construction of the root formatter, this is where they can do that.

## SOURCE

The source code repository for Test2 can be found at

<http://github.com/Test-More/test-more/>.

## MAINTAINERS

Chad Granum <[exodist@cpan.org](mailto:exodist@cpan.org)>

## AUTHORS

Chad Granum <[exodist@cpan.org](mailto:exodist@cpan.org)>

## COPYRIGHT

Copyright 2020 Chad Granum <[exodist@cpan.org](mailto:exodist@cpan.org)>.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

See <http://dev.perl.org/licenses/>

perl v5.34.0

2023-11-23

Test2::Formatter(3perl)