



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::EventFacet::Trace.3perl'

\$ man Test2::EventFacet::Trace.3perl

Test2::EventFacet::Trace(3perl) Perl Programmers Reference Guide Test2::EventFacet::Trace(3perl)

NAME

Test2::EventFacet::Trace - Debug information for events

DESCRIPTION

The Test2::API::Context object, as well as all Test2::Event types need to have access to information about where they were created. This object represents that information.

SYNOPSIS

```
use Test2::EventFacet::Trace;

my $trace = Test2::EventFacet::Trace->new(
    frame => [$package, $file, $line, $subname],
);
```

FACET FIELDS

\$string = \$trace->{details}

\$string = \$trace->details()

Used as a custom trace message that will be used INSTEAD of "at <FILE> line <LINE>" when calling "\$trace->debug".

\$frame = \$trace->{frame}

```
$frame = $trace->frame()
```

Get the call frame arrayref.

```
[$package, $file, $line, $subname]
```

```
$sint = $trace->{pid}
```

```
$sint = $trace->pid()
```

The process ID in which the event was generated.

```
$sint = $trace->{tid}
```

```
$sint = $trace->tid()
```

The thread ID in which the event was generated.

```
$sid = $trace->{cid}
```

```
$sid = $trace->cid()
```

The ID of the context that was used to create the event.

```
$suid = $trace->{uuid}
```

```
$suid = $trace->uuid()
```

The UUID of the context that was used to create the event. (If uuid tagging was enabled)

```
($pkg, $file, $line, $subname) = $trace->call
```

Get the basic call info as a list.

```
@caller = $trace->full_call
```

Get the full caller(N) results.

```
$warning_bits = $trace->warning_bits
```

Get index 9 from the full caller info. This is the warnings_bits field.

The value of this is not portable across perl versions or even processes. However it can be used in the process that generated it to reproduce the warnings settings in a

new scope.

```
eval <<EOT;  
BEGIN { ${^WARNING_BITS} = $trace->warning_bits };  
... context's warning settings apply here ...  
EOT
```

DISCOURAGED HUB RELATED FIELDS

These fields were not always set properly by tools. These are MOSTLY deprecated by the Test2::EventFacet::Hub facets. These fields are not required, and may only reflect the hub that was current when the event was created, which is not necessarily the same as the hub the event was sent through.

Some tools did do a good job setting these to the correct hub, but you cannot always rely on that. Use the 'hubs' facet list instead.

```
$hid = $trace->{hid}
```

```
$hid = $trace->hid()
```

The ID of the hub that was current when the event was created.

```
$huuid = $trace->{huuid}
```

```
$huuid = $trace->huuid()
```

The UUID of the hub that was current when the event was created. (If uuid tagging was enabled).

```
$int = $trace->{nested}
```

```
$int = $trace->nested()
```

How deeply nested the event is.

```
$bool = $trace->{buffered}
```

```
$bool = $trace->buffered()
```

True if the event was buffered and not sent to the formatter independent of a parent (This should never be set when nested is 0 or "undef").

METHODS

Note: All facet frames are also methods.

```
$trace->set_detail($msg)
```

```
$msg = $trace->detail
```

Used to get/set a custom trace message that will be used INSTEAD of "at <FILE> line <LINE>" when calling "\$trace->debug".

"detail()" is an alias to the "details" facet field for backwards compatibility.

```
$str = $trace->debug
```

Typically returns the string "at <FILE> line <LINE>". If "detail" is set then its value will be returned instead.

```
$trace->alert($MESSAGE)
```

This issues a warning at the frame (filename and line number where errors should be reported).

```
$trace->throw($MESSAGE)
```

This throws an exception at the frame (filename and line number where errors should be reported).

```
($package, $file, $line, $subname) = $trace->call()
```

Get the caller details for the debug-info. This is where errors should be reported.

```
$pkg = $trace->package
```

Get the debug-info package.

```
$file = $trace->file
```

Get the debug-info filename.

```
$line = $trace->line
```

Get the debug-info line number.

```
$subname = $trace->subname
```

Get the debug-info subroutine name.

```
$sig = trace->signature
```

Get a signature string that identifies this trace. This is used to check if multiple events are related. The signature includes pid, tid, file, line number, and the cid.

SOURCE

The source code repository for Test2 can be found at
<http://github.com/Test-More/test-more/>.

MAINTAINERS

Chad Granum <exodist@cpan.org>

AUTHORS

Chad Granum <exodist@cpan.org>

COPYRIGHT

Copyright 2020 Chad Granum <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/>