



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::Util::Facets2Legacy.3perl'

\$ man Test2::Util::Facets2Legacy.3perl

Test2::Util::Facets2Legacy(3perl)Perl Programmers Reference GuidTest2::Util::Facets2Legacy(3perl)

NAME

Test2::Util::Facets2Legacy - Convert facet data to the legacy event API.

DESCRIPTION

This module exports several subroutines from the older event API (see Test2::Event). These subroutines can be used as methods on any object that provides a custom "facet_data()" method. These subroutines can also be used as functions that take a facet data hashref as arguments.

SYNOPSIS

AS METHODS

```
package My::Event;

use Test2::Util::Facets2Legacy ':ALL';

sub facet_data { return { ... } }
```

Then to use it:

```
my $e = My::Event->new(...);
```

```
my $causes_fail = $e->causes_fail;
my $summary     = $e->summary;
....
```

AS FUNCTIONS

```
use Test2::Util::Facets2Legacy ':ALL';
```

```
my $f = {
    assert => { ... },
    info => [{...}, ...],
    control => {...},
    ...
};
```

```
my $causes_fail = causes_fail($f);
my $summary     = summary($f);
```

NOTE ON CYCLES

When used as methods, all these subroutines call "\$e->facet_data()". The default "facet_data()" method in Test2::Event relies on the legacy methods this module emulates in order to work. As a result of this it is very easy to create infinite recursion bugs.

These methods have cycle detection and will throw an exception early if a cycle is detected. "uuid()" is currently the only subroutine in this library that has a fallback behavior when cycles are detected.

EXPORTS

Nothing is exported by default. You must specify which methods to import, or use the ':ALL' tag.

```
$bool = $e->causes_fail()
```

```
$bool = causes_fail($f)
```

Check if the event or facets result in a failing state.

```
$bool = $e->diagnostics()
```

```
$bool = diagnostics($f)
```

Check if the event or facets contain any diagnostics information.

```
$bool = $e->global()
```

```
$bool = global($f)
```

Check if the event or facets need to be globally processed.

```
$bool = $e->increments_count()
```

```
$bool = increments_count($f)
```

Check if the event or facets make an assertion.

```
$bool = $e->no_display()
```

```
$bool = no_display($f)
```

Check if the event or facets should be rendered or hidden.

```
($max, $directive, $reason) = $e->sets_plan()
```

```
($max, $directive, $reason) = sets_plan($f)
```

Check if the event or facets set a plan, and return the plan details.

```
$id = $e->subtest_id()
```

```
$id = subtest_id($f)
```

Get the subtest id, if any.

```
$string = $e->summary()
```

```
$string = summary($f)
```

Get the summary of the event or facets hash, if any.

```
$undef_or_int = $e->terminate()
```

```
$undef_or_int = terminate($f)
```

Check if the event or facets should result in process termination, if so the exit code is returned (which could be 0). undef is returned if no termination is requested.

```
$uuid = $e->uuid()
```

```
$uuid = uuid($f)
```

Get the UUID of the facets or event.

Note: This will fall back to "\$e->SUPER::uuid()" if a cycle is detected and an event is used as the argument.

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/>