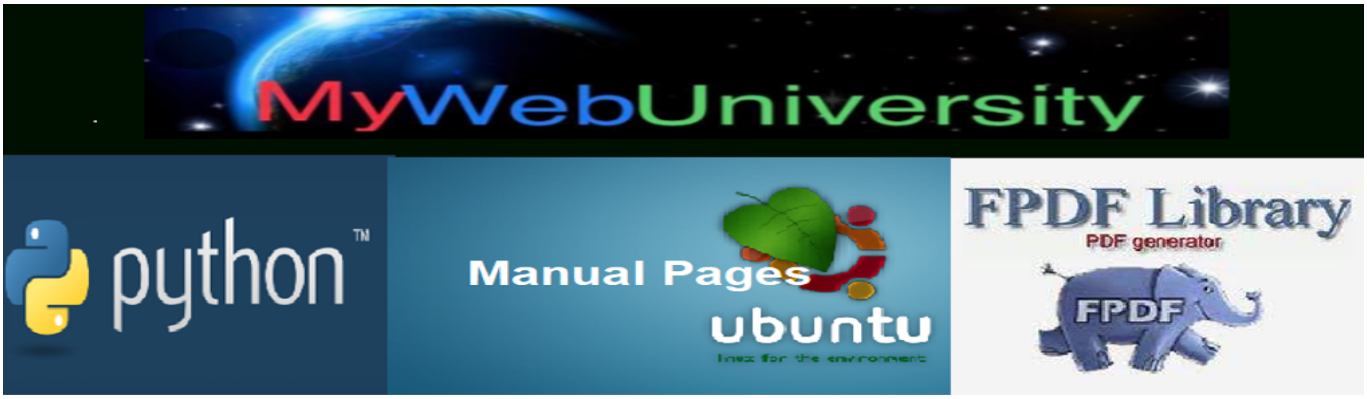

Rocky Enterprise Linux 9.2 Manual Pages on command 'TAP::Parser::SourceHandler::Executable.3perl'

$ man TAP::Parser::SourceHandler::Executable.3perl

TAP::Parser::SourceHandler::ExecuPerleProgrammers ReTAP::Parser::SourceHandler::Executable(3perl)

NAME

TAP::Parser::SourceHandler::Executable - Stream output from an executable TAP source

VERSION

Version 3.43

SYNOPSIS

```
use TAP::Parser::Source;
use TAP::Parser::SourceHandler::Executable;
my $source = TAP::Parser::Source->new->raw(['/usr/bin/ruby', 'mytest.rb']);
$source->assemble_meta;
my $class = 'TAP::Parser::SourceHandler::Executable';
my $vote  = $class->can_handle( $source );
my $iter  = $class->make_iterator( $source );
```

DESCRIPTION

This is an executable TAP::Parser::SourceHandler - it has 2 jobs:

1. Figure out if the TAP::Parser::Source it's given is an executable
   command ("can_handle").
2. Creates an iterator for executable commands ("make_iterator").

Unless you're writing a plugin or subclassing TAP::Parser, you probably won't need to use this module directly.

METHODS

Class Methods

"can_handle"

```
  my $vote = $class->can_handle( $source );
```

Only votes if $source looks like an executable file. Casts the following votes:

```
  0.9  if it's a hash with an 'exec' key
  0.8  if it's a .bat file
  0.75 if it's got an execute bit set
```

"make_iterator"

```
  my $iterator = $class->make_iterator( $source );
```

Returns a new TAP::Parser::Iterator::Process for the source. "$source->raw" must be in one of the following forms:

```
  { exec => [ @exec ] }
  [ @exec ]
  $file
```

"croak"s on error.

"iterator_class"

The class of iterator to use, override if you're sub-classing. Defaults to TAP::Parser::Iterator::Process.

SUBCLASSING

Please see "SUBCLASSING" in TAP::Parser for a subclassing overview.

Example

```
  package MyRubySourceHandler;
  use strict;
  use Carp qw( croak );
  use TAP::Parser::SourceHandler::Executable;
  use base 'TAP::Parser::SourceHandler::Executable';
  # expect $handler->(['mytest.rb', 'cmdline', 'args']);
  sub make_iterator {
    my ($self, $source) = @_;
    my @test_args = @{ $source->test_args };
    my $rb_file   = $test_args[0];
    croak("error: Ruby file '$rb_file' not found!") unless (-f $rb_file);
    return $self->SUPER::raw_source(['/usr/bin/ruby', @test_args]);
  }
```

SEE ALSO

TAP::Object, TAP::Parser, TAP::Parser::IteratorFactory, TAP::Parser::SourceHandler,

TAP::Parser::SourceHandler::Perl, TAP::Parser::SourceHandler::File,

TAP::Parser::SourceHandler::Handle, TAP::Parser::SourceHandler::RawTAP