NAME

TAP::Parser::Source::Perl - Stream Perl output

VERSION

Version 3.17

SYNOPSIS

```
use TAP::Parser::Source::Perl;
my $perl = TAP::Parser::Source::Perl->new;
my $stream = $perl->source( [ $filename, @args ] )->get_stream;
```

DESCRIPTION

Takes a filename and hopefully returns a stream from it. The filename should be the name of a Perl program.

Note that this is a subclass of *TAP::Parser::Source*. See that module for more methods.

METHODS

Class Methods

new

```
 my $perl = TAP::Parser::Source::Perl->new;
```

Returns a new `TAP::Parser::Source::Perl` object.

Instance Methods

source

Getter/setter the name of the test program and any arguments it requires.

```
my ($filename, @args) = @{ $perl->source };
$perl->source( [ $filename, @args ] );
```

`croak`s if `$filename` could not be found.

switches

```
my $switches = $perl->switches;
my @switches = $perl->switches;
$perl->switches( \@switches );
```

Getter/setter for the additional switches to pass to the perl executable. One common switch would be to set an include directory:

```
$perl->switches( ['-Ilib'] );
```

get_stream

```
my $stream = $source->get_stream($parser);
```

Returns a stream of the output generated by executing `source`. Must be passed an object that implements a `make_iterator` method. Typically this is a TAP::Parser instance.

shebang

Get the shebang line for a script file.

```
my $shebang = TAP::Parser::Source::Perl->shebang( $some_script );
```

May be called as a class method

get_taint

Decode any taint switches from a Perl shebang line.

```
# $taint will be 't'
my $taint = TAP::Parser::Source::Perl->get_taint( '#!/usr/bin/perl -t' );


# $untaint will be undefined
my $untaint = TAP::Parser::Source::Perl->get_taint( '#!/usr/bin/perl' );
```

SUBCLASSING

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

Example

```
package MyPerlSource;


use strict;
use vars '@ISA';


use Carp qw( croak );
use TAP::Parser::Source::Perl;


@ISA = qw( TAP::Parser::Source::Perl );


sub source {
    my ($self, $args) = @_;
    if ($args) {
 $self->{file} = $args->[0];
 return $self->SUPER::source($args);
    }
    return $self->SUPER::source;
}


# use the version of perl from the shebang line in the test file
sub _get_perl {
    my $self = shift;
    if (my $shebang = $self->shebang( $self->{file} )) {
        $shebang =~ /^#!(.*\bperl.*?)(?:(?:\s)|(?:$))/;
 return $1 if $1;
    }
    return $self->SUPER::_get_perl(@_);
}
```

SEE ALSO

TAP::Object, *TAP::Parser*, *TAP::Parser::Source*,