

NAME

TAP::Parser::SourceHandler::RawTAP - Stream output from raw TAP in a scalar/array ref.

VERSION

Version 3.36

SYNOPSIS

```
use TAP::Parser::Source;
use TAP::Parser::SourceHandler::RawTAP;

my $source = TAP::Parser::Source->new->raw( \"1..1\nok 1\n" );
$source->assemble_meta;

my $class = 'TAP::Parser::SourceHandler::RawTAP';
my $vote = $class->can_handle( $source );
my $iter = $class->make_iterator( $source );
```

DESCRIPTION

This is a *raw TAP output* [TAP::Parser::SourceHandler](#) - it has 2 jobs:

1. Figure out if the [TAP::Parser::Source](#) it's given is raw TAP output (“can_handle”).
2. Creates an iterator for raw TAP output (“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 is an array, or a scalar with newlines. Casts the following votes:

```
0.9 if it's a scalar with '..' in it
0.7 if it's a scalar with 'ok' in it
0.3 if it's just a scalar with newlines
0.5 if it's an array
```

make_iterator

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

Returns a new [TAP::Parser::Iterator::Array](#) for the source. \$source->raw must be an array ref, or a scalar ref.

croaks on error.

SUBCLASSING

Please see “SUBCLASSING” in [TAP::Parser](#) for a subclassing overview.

SEE ALSO

[TAP::Object](#), [TAP::Parser](#), [TAP::Parser::IteratorFactory](#), [TAP::Parser::SourceHandler](#),
[TAP::Parser::SourceHandler::Executable](#), [TAP::Parser::SourceHandler::Perl](#),
[TAP::Parser::SourceHandler::File](#), [TAP::Parser::SourceHandler::Handle](#)