

NAME

TAP::Parser::SourceHandler::Handle - Stream TAP from an IO::Handle or a GLOB.

VERSION

Version 3.36

SYNOPSIS

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

my $source = TAP::Parser::Source->new->raw( \*TAP_FILE );
$source->assemble_meta;

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

DESCRIPTION

This is a *raw TAP stored in an IO Handle* [TAP::Parser::SourceHandler](#) class. It has 2 jobs:

1. Figure out if the [TAP::Parser::Source](#) it's given is an [IO::Handle](#) or GLOB containing raw TAP output (“can_handle”).
2. Creates an iterator for IO::Handle's & globs (“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 );
```

Casts the following votes:

```
0.9 if $source is an IO::Handle
0.8 if $source is a glob
```

make_iterator

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

Returns a new [TAP::Parser::Iterator::Stream](#) for the source.

iterator_class

The class of iterator to use, override if you're sub-classing. Defaults to [TAP::Parser::Iterator::Stream](#).

SUBCLASSING

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

SEE ALSO

[TAP::Object](#), [TAP::Parser](#), [TAP::Parser::Iterator](#), [TAP::Parser::Iterator::Stream](#),
[TAP::Parser::IteratorFactory](#), [TAP::Parser::SourceHandler](#), [TAP::Parser::SourceHandler::Executable](#),
[TAP::Parser::SourceHandler::Perl](#), [TAP::Parser::SourceHandler::File](#),
[TAP::Parser::SourceHandler::RawTAP](#)