

## NAME

TAP::Parser::Iterator::Process - Internal TAP::Parser Iterator

## VERSION

Version 3.17

## SYNOPSIS

```
# see TAP::Parser::IteratorFactory for preferred usage

# to use directly:
use TAP::Parser::Iterator::Process;
my %args = (
    command => ['python', 'setup.py', 'test'],
    merge   => 1,
    setup   => sub { ... },
    teardown => sub { ... },
);
my $it    = TAP::Parser::Iterator::Process->new(\%args);
my $line = $it->next;
```

## DESCRIPTION

This is a simple iterator wrapper for executing external processes, used by *TAP::Parser*. Unless you're subclassing, you probably won't need to use this module directly.

## METHODS

### Class Methods

#### new

Create an iterator. Expects one argument containing a hashref of the form:

```
command => \@command_to_execute
merge   => $attempt_merge_stderr_and_stdout?
setup   => $callback_to_setup_command
teardown => $callback_to_teardown_command
```

Tries to uses *IPC::Open3* & *IO::Select* to communicate with the spawned process if they are available. Falls back onto `open()`.

### Instance Methods

#### next

Iterate through the process output, of course.

#### next\_raw

Iterate raw input without applying any fixes for quirky input syntax.

#### wait

Get the wait status for this iterator's process.

#### exit

Get the exit status for this iterator's process.

#### handle\_unicode

Upgrade the input stream to handle UTF8.

**get\_select\_handles**

Return a list of filehandles that may be used upstream in a `select()` call to signal that this Iterator is ready. Iterators that are not handle based should return an empty list.

**ATTRIBUTION**

Originally ripped off from *Test::Harness*.

**SEE ALSO**

*TAP::Object*, *TAP::Parser*, *TAP::Parser::Iterator*, *TAP::Parser::IteratorFactory*,