



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'TAP::Parser::Result::Test.3perl'

\$ man TAP::Parser::Result::Test.3perl

TAP::Parser::Result::Test(3perl) Perl Programmers Reference GuideTAP::Parser::Result::Test(3perl)

NAME

TAP::Parser::Result::Test - Test result token.

VERSION

Version 3.43

DESCRIPTION

This is a subclass of TAP::Parser::Result. A token of this class will be returned if a test line is encountered.

1..1

ok 1 - woo hooo!

OVERRIDDEN METHODS

This class is the workhorse of the TAP::Parser system. Most TAP lines will be test lines and if "\$result->is_test", then you have a bunch of methods at your disposal.

Instance Methods

"ok"

my \$ok = \$result->ok;

Returns the literal text of the "ok" or "not ok" status.

"number"

```
my $test_number = $result->number;
```

Returns the number of the test, even if the original TAP output did not supply that number.

"description"

```
my $description = $result->description;
```

Returns the description of the test, if any. This is the portion after the test number but before the directive.

"directive"

```
my $directive = $result->directive;
```

Returns either "TODO" or "SKIP" if either directive was present for a test line.

"explanation"

```
my $explanation = $result->explanation;
```

If a test had either a "TODO" or "SKIP" directive, this method will return the accompanying explanation, if present.

```
not ok 17 - 'Pigs can fly' # TODO not enough acid
```

For the above line, the explanation is not enough acid.

"is_ok"

```
if ( $result->is_ok ) { ... }
```

Returns a boolean value indicating whether or not the test passed. Remember that for TODO tests, the test always passes.

If the test is unplanned, this method will always return false. See "is_unplanned".

"is_actual_ok"

```
if ( $result->is_actual_ok ) { ... }
```

Returns a boolean value indicating whether or not the test passed, regardless of its TODO status.

"actual_passed"

Deprecated. Please use "is_actual_ok" instead.

"todo_passed"

```
if ( $test->todo_passed ) {  
    # test unexpectedly succeeded  
}
```

If this is a TODO test and an 'ok' line, this method returns true. Otherwise, it will always return false (regardless of passing status on non-todo tests).

This is used to track which tests unexpectedly succeeded.

"todo_failed"

deprecated in favor of 'todo_passed'. This method was horribly misnamed.

This was a badly misnamed method. It indicates which TODO tests unexpectedly succeeded.

Will now issue a warning and call "todo_passed".

"has_skip"

```
if ( $result->has_skip ) { ... }
```

Returns a boolean value indicating whether or not this test has a SKIP directive.

"has_todo"

```
if ( $result->has_todo ) { ... }
```

Returns a boolean value indicating whether or not this test has a TODO directive.

"as_string"

```
print $result->as_string;
```

This method prints the test as a string. It will probably be similar, but not necessarily identical, to the original test line. Directives are capitalized, some whitespace may be trimmed and a test number will be added if it was not present in the original line. If you need the original text of the test line, use the "raw" method.

"is_unplanned"

```
if ( $test->is_unplanned ) { ... }
```

```
$test->is_unplanned(1);
```

If a test number is greater than the number of planned tests, this method will return

true. Unplanned tests will always return false for "is_ok", regardless of whether or not the test "has_todo".

Note that if tests have a trailing plan, it is not possible to set this property for unplanned tests as we do not know it's unplanned until the plan is reached:

```
print <<'END';
```

```
ok 1
```

```
ok 2
```

```
1..1
```

```
END
```

```
perl v5.34.0
```

```
2023-11-23
```

```
TAP::Parser::Result::Test(3perl)
```