

#### 15.4.3.2.4 Test Anything Protocol (TAP) *Test Driver*

The Test Anything Protocol (TAP) is a Error: Reference source not found. The protocol TAP Standard is defined at <http://testanything.org/> as modified by Automake. The TAP *Test Driver* is called a TAP Harness in the TAP Standard. YAML Standard (<http://www.yaml.org/spec/1.2/spec.html>)

TAP requires an interfacing *test class* program to output ASCII lines as defined by the TAP Protocol. The TAP test driver analyses these lines and outputs a log file and Test Results File compatible with the Error: Reference source not found.

##### 15.4.3.2.4.1 TAP Protocol

A *test case* using the TAP protocol must output formatted lines of ASCII text conforming the TAP Standard, as modified by Automake. Each line must start in column 1 except for YAML comments which must start in column 2 or more.

A little terminology:

- The *test case* is the name of the script of program generating the TAP Protocol compliant data.
- A *test case* executes tests. The result of each test execution is a TAP formatted test line, YAML comments and Diagnostics.
- Whitespace <ws> is either a blank, ' ', or a tab character, \t.
- Where one <ws> is required, many can be used.
- A test line is any line which has an **ok** or **not ok** in column 1.
- All test lines are sequentially numbered, during processing, starting at one (1). The current test line number is the current sequential number.

The format of TAP Protocol Element is:

TAP Type	Syntax	Description
Test plan	[1..N]	N the number of tests.
Test plan	1..0 [# comment]	Skip tests
Test line	ok [N] [comment] [# Directive]	Test success (or skip)
Test line	not ok [N] [comment]	Test fail.
Bail out!	Bail out! [comment]	Terminate processing.
Diagnostic	# comment	A free from comment.
YAML	--- ...	Comments bracketed by --- and ...

The description of the TAP Protocol Elements is:

	The test plan is required and must precede the first test line or follow the last test line. Diagnostics are optional before and after the test plan.
<b>1..N</b>	The number <b>N</b> must be the number of test lines.
<b>1..0</b>	Indicates that the test case is skipped.
<b>ok</b>	Test success. This is case sensitive.
<b>not ok</b>	Test failure, This is case sensitive. There must be a single blank character between <b>not</b> and <b>ok</b> .
<b>N</b>	Test number. Must be the same as the test line number if in a test line.
<b>comment</b>	Free form comment not containing a hash (#).
<b>Directive</b>	The first word must be either <b>SKIP</b> or <b>TODO</b> . The remainder of the line is treated as a free form comment. <b>SKIP</b> and <b>TODO</b> . are not case sensitive.
<b>Bail out!</b>	Stop processing the current <i>test case</i> TAP Protocol.
<b>Diagnostic</b>	A line beginning with a '#' and followed by a <b>comment</b> .
<b>YAML</b>	<b>YAML</b> lines are a comments and are ignored for processing.

The requirements for satisfying the TAP Protocol are:

- By definition the a test line number is formed by incrementing a test line number after each test line with the first test line being one (1). The current test line number is the value of the current number. Note that non-test lines are not counted.
- Test Plans:
  - A Test Plan must be the first non-Diagnostic line or the last line before 0 or more Diagnostic lines, that is, the Test Plan must follow all test lines and **YAML** text.
  - If there is a test plan, **1..N**, where **N** is greater than zero, then if the number of test lines is less than **N**, say **k** is the last test line, then each **k+1 .. N** missing tests line will be considered as **SKIPed**. A diagnostic message will be logged.
  - If there is a test plan, **1..N**, where **N** is greater than zero, hen if the number of test lines is greater than **N**, say **k** is the last test line, then **N+1 .. k** test lines will be accepted and a diagnostic message will be logged.
  - If there is a test plan, **1..0** then the *test case* will be considered as **SKIPed** and no test lines are required. If any test lines are present then they will be accepted but but the test case will be considered as **SKIPed**.
- Test Lines:
  - **ok** and **not ok** are case sensitive.
  - If a test line has a number, **N**, it must be separated from the preceding **ok** by <ws>. The number **N** must be the current test line number.
  - The test line **not ok** must have a single blank character between **not** and **ok**.
  - If the test line has a **comment**, then the **comment** must be separated from the preceding **ok** or **N** by <ws>. A **comment** can not have a leading number or an embedded hash character..

- The Directive separator is a hash sign '#'.
- A Directive must be **SKIP** or **TODO**. Directives are case insensitive. **SKIP** has the same meaning as in Error: Reference source not found.
- **Bail out!:** Must start in column 1 and there must be a single space between **Bail** and **out!**. Any comments must have <ws> between **out!** and the start of the comment. No further tests will be processed for the current *test case*.
- **Diagnostic:** If the first character seen is a hash character, '#', then the line is a comment. There is no comment processing.
- **YAML** comments are comments and are ignored. **YAML** comments can not start in column 1, this includes the bracketing --- and .... The **YAML** brackets must be in a solitary line.
- All input will be logged into a .log file.
- All input which does not satisfy the requirements will be flagged as an error.

A sample of *test case* TAP Protocol output is:

```
test line
number N
    1..55
    1    ok
    2    ok 2
        # this is a comment
    3    not ok 17 comment
    4    ok - comment # skip skip reason
        ---
        This is a YAML comment.
        It can be extend across multiple lines.
        '---' and '...' must be on their own line.
        ...
```

And:

- **1..55** is the test plan. The *test case* output is in error because there are only 4 tests. The test plan is optional and can appear as the first or last physical line in the *test case* output.
- Test 3 is in error because N is 17 and not 3.
- Test 4 is a skipped test.
- The **YAML** comments do not start in column 1 and are multiline.

#### 15.4.3.2.4.2 TAP Makefile.am Variables

##### Automake – TAP Variables

<b>AM_TAP_AWK</b>	Puts a reference to the local AWK into the environment.
-------------------	---

This must be used in the **ext\_LOG\_DRIVER** variable to initiate TAP Protocol processing. The only acceptable way to reference the TAP Protocol processor is:

```
TEST_LOG_DRIVER = env AM_TAP_AWK='$(AWK)' $(SHELL) \
    $(top_srcdir)/build-aux/tap-driver.sh
```

#### 15.4.3.2.4.3 *Test Harness to TAP Test Driver Interface*

Section Error: Reference source not found plus the following *Test Harness to TAP Test Driver Options*. The options are input to the TAP driver using the `ext_DRIVER_FLAGS` or `AM_ext_DRIVER_FLAGS` variables, as in `ext_DRIVER_FLAGS = --comments`.

The `--enable-hard-errors` is not supported in the TAP Protocol and is ignored.

#### *Test Harness to TAP Test Driver Options*

<code>--long_form=value</code>	description
<code>--comments</code>	Display Diagnostics to the System Administrator monitor.
<code>--diagnostic-string</code>	Change the Diagnostic prefix, '#', with a string of 1 or more characters.
<code>--ignore-exit</code>	Causes the test driver to ignore the exit status of the test scripts; by default, the driver will report an error if the script exits with a non-zero status. This option has effect also on non-zero exit statuses due to termination by a signal.
<code>--merge</code>	Instruct the test driver to merge the test class standard error into the Test Driver standard output.
<code>--no-comments</code>	Default: Do not display Diagnostics to the System Administrator monitor.
<code>--no-merge</code>	Default: Standard error and standard output are not merged.

#### 15.4.3.2.4.4 *TAP Test Driver to Test Harness Interface*

Results returned are compatible with the Test Results File. The results can be divided into local, dealing with each test line, and global, dealing with the test case. The global results are defined in the Test Results File values of `:recheck:` and `:test-global-result:` and `:copy-in-global-log:`. The local results are captured in `:test-result:`.

A *test case* can have 1 or more tests. The result of each test is a TAP test line. If the test plan is **1..0** then there can be no tests.

In all cases **PASS** is substituted for **ok** and **FAIL** is substituted for **not ok** in the following discussions.

The formation rules to generate these results are:

- **:test-result:** Either **PASS**, **FAIL** or **SKIP** as determined by each test line. The expected Test Results File (.trs) output is given in the following table.

options	TEST	.trs output
<b>expect</b>		<b>:test-result:</b>
--	PASS	PASS
no	FAIL	FAIL
yes	FAIL	
yes	PASS	
--	SKIP	SKIP

- The global result is the computed cumulative value for all test lines. The global result is considered the *test case* final value. The results are calculated in the following way:
  - If any test line is **not ok** then the global result is **FAIL**.
  - Test lines with **SKIP** are considered as **PASS** when calculating the global result. If all test lines are **SKIP**ed then the test case is considered as **SKIP**ed.
  - For a test plan of **1..N**, if the number of test lines is  $< N$  then the missing tests are marked as **SKIP** and **:test-result: SKIP** is output.
  - For a test plan of **1..N**, if the number of test lines is  $> N$  then **N** will be considered as a incorrect number, the test lines will be accepted as if  $N + k$ , where **k** is the excess test lines over **N**, then it will be assumed that  $1..N + k$  was given as the test plan..
  - For a test plan of **1..0** the global result will be **SKIP**. **SKIP** trumps **FAIL**. If test lines are included and if any test line is **not ok** the global test result will be **SKIP** not **FAIL**.
  - If all test results are **SKIP** then the global result will be **SKIP**, otherwise a single **XPASS** will cause the global result to be reported as a **FAIL**.
  - In summary

$\Sigma$ test lines	.trs			exit
	result	:recheck:	copy	
PASS	PASS	no	no	0
FAIL	FAIL	yes	yes	2
SKIP	SKIP	no	yes	77

where

$\Sigma$ test lines    global result calculation

result            :test-global-result:

:recheck:        .recheck:

copy             :copy-in-global:

exit              tap-driver.sh return value

#### **15.4.3.2.4.5      *Test Class to TAP Test Driver Interface***

This is defined as the TAP Protocol in TAP Protocol.

#### **15.4.4      *Parallel Test Cases***

##### **15.4.4.2.4 *Test Case Preprocessing***

##### **15.4.4.3      *Test Driver to Test Case Interfaces***

##### **15.4.4.4      *Test Case to Test Driver Interface***

#### **15.4.5      *dejaGnu Test Harness***

#### **15.5      *Use of make***

#### **15.6      *Examples***