9pm

Test Report

9pm 32e813c

2025-09-14

Table of Contents

1. Test Summary

1.1. 9pm Info

Version	32e813c
SHA	32e813c14b1b

1.2. Test Overview

PASS	40
FAIL	0
SKIP	6
~~FAIL~~	2
~~SKIP~~	4
TOTAL	**52**

- **SKIP** auto.yaml
  - **PASS** dummy.sh
  - **PASS** noopt.tcl
  - **SKIP** Options
    - **PASS** options.tcl
    - **SKIP** Argument Skip
    - **PASS** A Nested Suite
      - **PASS** options.tcl
  - **SKIP** Suite Skip
    - **SKIP** suite_skip.tcl
    - **~~SKIP~~** suite_skip.tcl
    - **SKIP** suite_skip.tcl
  - **PASS** Test Report Creation
    - **PASS** test.pl
    - **PASS** Verify fixed test-spec: Readme.adoc
    - **PASS** dynamic.yaml
      - **PASS** bash.sh
      - **PASS** Verify dynamic test-spec: <case>.adoc
  - **PASS** On Fail
    - **~~FAIL~~** onfail_fail.tcl
    - **PASS** onfail_check.tcl

- **PASS** Lib TCL
  - **PASS** Args
    - **PASS** noopt.sh
    - **PASS** noopt.tcl
    - **PASS** precedence.sh
  - **PASS** Config
    - **PASS** config.tcl
  - **PASS** Execute
    - **PASS** execute.tcl
  - **PASS** Logging
    - **PASS** logging.tcl
  - **PASS** Shell
    - **PASS** shell_active.tcl
    - **PASS** shell_stack.tcl
    - **PASS** shell_test.tcl
  - **PASS** Helpers
    - **PASS** retry_test.tcl
    - **PASS** root_path.tcl
    - **PASS** getopts.tcl
  - **PASS** Database
    - **PASS** db_scratch.tcl
    - **PASS** db_write.tcl
    - **PASS** db_read.tcl
  - **PASS** TAP
    - **~~SKIP~~** OK Skip
    - **~~SKIP~~** OK Skip OK
    - **~~SKIP~~** Skip
    - **PASS** OK
    - **PASS** Plan First
    - **PASS** Plan Last
    - **~~FAIL~~** Plan Omit

2. Test Result

2.1. PASS dummy.sh

Output

```
Starting: unit_tests/sanity/dummy.sh cmdl-supplied

2025-09-14 07:03:39 0..1
2025-09-14 07:03:39 ok 1 - Looks OK to me
```

2.2. PASS noopt.tcl

Output

```
Starting: unit_tests/sanity/noopt.tcl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - No options passed by default (only "cmdl-supplied")
```

2.3. PASS options.tcl

Output

```
Starting: unit_tests/harness/options.tcl suite-supplied mode:long optional-positive:yes
                                         skip-positive:yes scratch:/tmp/9pm_i78hxc2f
                                         base:/home/jocke/src/infix/test/9pm/unit_tests/harness
                                         cmdl-supplied

2025-09-14 07:03:39 1..6
2025-09-14 07:03:39 ok 1 - correct number of arguments
2025-09-14 07:03:39 ok 2 - argument from suite
2025-09-14 07:03:39 ok 3 - argument from command line
2025-09-14 07:03:39 ok 4 - got optional argument
2025-09-14 07:03:39 # INFO - got scratch option = /tmp/9pm_i78hxc2f
2025-09-14 07:03:39 ok 5 - <scratch> in suite evaluated to SCRATCHDIR
2025-09-14 07:03:39 # INFO - got base option = /home/jocke/src/infix/test/9pm/unit_tests/harness
2025-09-14 07:03:39 ok 6 - <base> in suite evaluated to some path
```

2.4. SKIP Argument Skip

Output

```
Starting: unit_tests/harness/options.tcl suite-supplied mode:long optional-positive:yes
                                         scratch:/tmp/9pm_i78hxc2f
                                         base:/home/jocke/src/infix/test/9pm/unit_tests/harness
                                         skip-positive:no cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 # skip Required argument "skip-positive" != "yes"
```

2.5. PASS options.tcl

Output

```
Starting: unit_tests/harness/options.tcl suite-supplied mode:long scratch:/tmp/9pm_i78hxc2f
                                         base:/home/jocke/src/infix/test/9pm/unit_tests/harness
                                         optional-positive:yes skip-positive:yes cmdl-supplied

2025-09-14 07:03:39 1..6
2025-09-14 07:03:39 ok 1 - correct number of arguments
2025-09-14 07:03:39 ok 2 - argument from suite
2025-09-14 07:03:39 ok 3 - argument from command line
2025-09-14 07:03:39 ok 4 - got optional argument
2025-09-14 07:03:39 # INFO - got scratch option = /tmp/9pm_i78hxc2f
2025-09-14 07:03:39 ok 5 - <scratch> in suite evaluated to SCRATCHDIR
2025-09-14 07:03:39 # INFO - got base option = /home/jocke/src/infix/test/9pm/unit_tests/harness
2025-09-14 07:03:39 ok 6 - <base> in suite evaluated to some path
```

2.6. SKIP suite_skip.tcl

Output

```
Starting: unit_tests/harness/suite_skip.tcl require-value-foo:bar cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 # skip suite Required argument "require-value-foo" != "foo"
```

2.7. PASS test.pl

Description

An example of how to create a 9pm test case by printing TAP from Perl.

This test specification is named Readme.adoc and is located in the same directory as test.pl.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sed auctor ipsum. Sed pharetra vestibulum leo, ac vulputate odio tristique eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed dictum, est ut elementum bibendum, velit odio feugiat ante, et posuere metus erat a nulla. Curabitur lacinia vel lacus in feugiat. Integer pulvinar vestibulum felis et vehicula. Sed facilisis turpis eget felis cursus feugiat. Quisque sodales mauris id sapien viverra, in suscipit arcu aliquet. Vestibulum porttitor aliquam arcu mattis efficitur. In hac habitasse platea dictumst. Vivamus aliquet pretium dolor, ut imperdiet ligula tristique vel. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Topology

Localhost only.

Test sequence

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Sed do eiusmod tempor incididunt
4. Ut labore et dolore magna aliqua
5. Excepteur sint occaecat cupidatat non proident

Output

```
Starting: unit_tests/harness/report/test.pl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - A good result from Perl
```

2.8. PASS Verify fixed test-spec: Readme.adoc

Description

An example of how to create a 9pm test case by printing TAP from Perl.

This test specification is named Readme.adoc and is located in the same directory as test.pl.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sed auctor ipsum. Sed pharetra vestibulum leo, ac vulputate odio tristique eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed dictum, est ut elementum bibendum, velit odio feugiat ante, et posuere metus erat a nulla. Curabitur lacinia vel lacus in feugiat. Integer pulvinar vestibulum felis et vehicula. Sed facilisis turpis eget felis cursus feugiat. Quisque sodales mauris id sapien viverra, in suscipit arcu aliquet. Vestibulum porttitor aliquam arcu mattis efficitur. In hac habitasse platea dictumst. Vivamus aliquet pretium dolor, ut imperdiet ligula tristique vel. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Topology

Localhost only.

Test sequence

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Sed do eiusmod tempor incididunt
4. Ut labore et dolore magna aliqua
5. Excepteur sint occaecat cupidatat non proident

Output

```
Starting: unit_tests/harness/report/test.pl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - A good result from Perl
```

2.9. PASS bash.sh

Description

An example of how to create a 9pm test case by printing TAP from Bash.

This test specification has the same name as the testcase but with the .adoc extension.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sed auctor ipsum. Sed pharetra vestibulum leo, ac vulputate odio tristique eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed dictum, est ut elementum bibendum, velit odio feugiat ante, et posuere metus erat a nulla. Curabitur lacinia vel lacus in feugiat. Integer pulvinar vestibulum felis et vehicula. Sed facilisis turpis eget felis cursus feugiat. Quisque sodales mauris id sapien viverra, in suscipit arcu aliquet. Vestibulum porttitor aliquam arcu mattis efficitur. In hac habitasse platea dictumst. Vivamus aliquet pretium dolor, ut imperdiet ligula tristique vel. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Topology

Localhost only.

Test sequence

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Sed do eiusmod tempor incididunt
4. Ut labore et dolore magna aliqua
5. Excepteur sint occaecat cupidatat non proident

Output

```
Starting: unit_tests/harness/report/dynamic/bash.sh cmdl-supplied

2025-09-14 07:03:39 1..2
2025-09-14 07:03:39 Got options: cmdl-supplied
2025-09-14 07:03:39 ok 1 - True is indeed true
2025-09-14 07:03:39 ok 2 - One is also true
```

2.10. PASS Verify dynamic test-spec: <case>.adoc

Description

An example of how to create a 9pm test case by printing TAP from Bash.

This test specification has the same name as the testcase but with the .adoc extension.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sed auctor ipsum. Sed pharetra vestibulum leo, ac vulputate odio tristique eget. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed dictum, est ut elementum bibendum, velit odio feugiat ante, et posuere metus erat a nulla. Curabitur lacinia vel lacus in feugiat. Integer pulvinar vestibulum felis et vehicula. Sed facilisis turpis eget felis cursus feugiat. Quisque sodales mauris id sapien viverra, in suscipit arcu aliquet. Vestibulum porttitor aliquam arcu mattis efficitur. In hac habitasse platea dictumst. Vivamus aliquet pretium dolor, ut imperdiet ligula tristique vel. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Topology

Localhost only.

Test sequence

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Sed do eiusmod tempor incididunt
4. Ut labore et dolore magna aliqua
5. Excepteur sint occaecat cupidatat non proident

Output

```
Starting: unit_tests/harness/report/dynamic/bash.sh cmdl-supplied

2025-09-14 07:03:39 1..2
2025-09-14 07:03:39 Got options: cmdl-supplied
2025-09-14 07:03:39 ok 1 - True is indeed true
2025-09-14 07:03:39 ok 2 - One is also true
```

2.11. ~~FAIL~~ onfail_fail.tcl

Output

```
Starting: unit_tests/harness/onfail_fail.tcl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 not ok 1 - I will always fail you
```

2.12. PASS onfail_check.tcl

Output

```
Starting: unit_tests/harness/onfail_check.tcl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 # INFO - Checking if onfail script wrote to database
2025-09-14 07:03:39 ok 1 - Database content checks out, onfail as executed
```

2.13. PASS noopt.sh

Output

```
Starting: unit_tests/lib_tcl/args/noopt.sh cmdl-supplied

2025-09-14 07:03:39 0..1
2025-09-14 07:03:39 ok 1 - No options passed by default (only cmdl-supplied)
```

2.14. PASS noopt.tcl

Output

```
Starting: unit_tests/lib_tcl/args/noopt.tcl cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - No options passed by default (only cmdl-supplied)
```

2.15. PASS precedence.sh

Output

```
Starting: unit_tests/lib_tcl/args/precedence.sh cmdl-supplied

2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - Got config passed from environment
2025-09-14 07:03:39 1..1
2025-09-14 07:03:39 ok 1 - Got config passed from cmdl
```

2.16. PASS config.tcl

Output

```
Starting: unit_tests/lib_tcl/config/config.tcl -c
                                               /home/jocke/src/infix/test/9pm/unit_tests/lib_tcl/config/config.yaml
                                               cmdl-supplied

2025-09-14 07:03:39 1..2
2025-09-14 07:03:39 ok 1 - Checking the integrity for the parsed configuration
2025-09-14 07:03:39 ok 2 - Iterating all node configurations and testing: get_node_info
```

2.17. PASS execute.tcl

Output

```
Starting: unit_tests/lib_tcl/execute/execute.tcl cmdl-supplied

2025-09-14 07:03:39 1..10
2025-09-14 07:03:39 # INFO - Checking line count for execute
2025-09-14 07:03:39 ok 1 - Execute line count
2025-09-14 07:03:39 # INFO - Checking line count for start, capture and finish
2025-09-14 07:03:39 ok 2 - Start, capture and finish line count
2025-09-14 07:03:39 # INFO - Testing user expect block releasing
2025-09-14 07:03:39 ok 3 - User expect block release
2025-09-14 07:03:39 # INFO - Testing command nesting
2025-09-14 07:03:39 ok 4 - Execute line count for nested command
2025-09-14 07:03:39 ok 5 - Got zero return for "ls /"
2025-09-14 07:03:39 ok 6 - Execute return code variable $? has sane values
2025-09-14 07:03:39 # INFO - Testing fast looped execution (1000 times)
2025-09-14 07:03:39 ok 7 - Command executed 1000 times
2025-09-14 07:03:39 # INFO - Testing command abort (100 times)
2025-09-14 07:03:41 ok 8 - Sleep command aborted 100
2025-09-14 07:03:41 # INFO - Testing command discard
2025-09-14 07:03:41 ok 9 - Started command discarded
2025-09-14 07:03:41 # INFO - Testing simultaneous commands that terminates
2025-09-14 07:03:41 ok 10 - Simultaneous commands on different shells (expect_after)
```

2.18. PASS logging.tcl

Output

```
Starting: unit_tests/lib_tcl/logging/logging.tcl cmdl-supplied

2025-09-14 07:03:41 1..3
2025-09-14 07:03:41 # INFO - efhchepjsnfxwtfhsvfj
2025-09-14 07:03:41 ok 1 - Logifile: /home/jocke/.local/share/9pm/logs/2025-09-14_07-03-39-170528/logging.tcl/0001/run.log
2025-09-14 07:03:41 ok 2 - Logifile: /home/jocke/.local/share/9pm/logs/2025-09-14_07-03-39-170528/last/run.log
2025-09-14 07:03:41 ok 3 - Logifile: /home/jocke/.local/share/9pm/logs/2025-09-14_07-03-39-170528/logging.tcl/last/run.log
```

2.19. PASS shell_active.tcl

Output

```
Starting: unit_tests/lib_tcl/shell/shell_active.tcl cmdl-supplied

2025-09-14 07:03:41 1..8
2025-09-14 07:03:41 # INFO - Spawn 3 new shells and verify active follows
2025-09-14 07:03:41 ok 1 - Active check for base
2025-09-14 07:03:41 ok 2 - Active check for sub1
2025-09-14 07:03:41 ok 3 - Active check for sub2
2025-09-14 07:03:41 # INFO - Switch to an existing shell and verify active follow
2025-09-14 07:03:41 ok 4 - Active check for sub1
2025-09-14 07:03:41 # INFO - Close active shell and verify active is unset
2025-09-14 07:03:41 ok 5 - No active shell
2025-09-14 07:03:41 # INFO - Switch to an existing shell then close other shell and verify active stays
2025-09-14 07:03:41 ok 6 - Active check for base
2025-09-14 07:03:41 ok 7 - Active check for base
2025-09-14 07:03:41 # INFO - Close shell with no active shell set
2025-09-14 07:03:41 ok 8 - Can close shell when no active shell is set
```

2.20. PASS shell_stack.tcl

Output

```
Starting: unit_tests/lib_tcl/shell/shell_stack.tcl cmdl-supplied

2025-09-14 07:03:41 1..8
2025-09-14 07:03:41 ok 1 - Base check for base
2025-09-14 07:03:41 ok 2 - Base check for subshell0
2025-09-14 07:03:41 ok 3 - Base check for subshell1
2025-09-14 07:03:41 ok 4 - Base check for subshell2
2025-09-14 07:03:41 ok 5 - Base check for subshell3
2025-09-14 07:03:41 ok 6 - Base check for subshell4
2025-09-14 07:03:41 ok 7 - Base check for subshell5
2025-09-14 07:03:41 ok 8 - Base check for base
```

2.21. PASS shell_test.tcl

Output

```
Starting: unit_tests/lib_tcl/shell/shell_test.tcl cmdl-supplied

2025-09-14 07:03:41 1..3
2025-09-14 07:03:41 # INFO - Creating 10 shells before closing them
2025-09-14 07:03:41 ok 1 - Having 10 shells open at one time
2025-09-14 07:03:41 # INFO - Creating 10 again, reusing the old names
2025-09-14 07:03:41 ok 2 - Reusing all 10 shells
2025-09-14 07:03:41 # INFO - Trying to swap back to all 10 open shells again
2025-09-14 07:03:41 ok 3 - Swapping back to all 10 open shells
```

2.22. PASS retry_test.tcl

Output

```
Starting: unit_tests/lib_tcl/helpers/retry_test.tcl cmdl-supplied

2025-09-14 07:03:41 1..6
2025-09-14 07:03:41 ok 1 - Count to 1 in 5 cycles
2025-09-14 07:03:44 ok 2 - Count to 4 in 5 cycles
2025-09-14 07:03:49 ok 3 - Fail to count to 9 in 5 cycles
2025-09-14 07:03:52 ok 4 - Run 3 seconds in 5 cycles and 1 second delay
2025-09-14 07:04:02 ok 5 - Run 10 seconds in 3 cycles and 5 second delay
2025-09-14 07:04:07 ok 6 - Fail to run 10 seconds in 5 cycles and 1 second delay
```

2.23. PASS root_path.tcl

Output

```
Starting: unit_tests/lib_tcl/helpers/root_path.tcl cmdl-supplied

2025-09-14 07:04:07 1..2
2025-09-14 07:04:07 ok 1 - $9pm::root_path exists
2025-09-14 07:04:07 ok 2 - Root path looks sane
```

2.24. PASS getopts.tcl

Output

```
Starting: unit_tests/lib_tcl/helpers/getopts.tcl cmdl-supplied

2025-09-14 07:04:07 1..3
2025-09-14 07:04:07 ok 1 - Bool test: force is set to true
2025-09-14 07:04:07 ok 2 - Value test: timeout set correctly
2025-09-14 07:04:07 ok 3 - Default value test: set correctly
```

2.25. PASS db_scratch.tcl

Output

```
Starting: unit_tests/lib_tcl/db/db_scratch.tcl cmdl-supplied

2025-09-14 07:04:07 1..2
2025-09-14 07:04:07 # INFO - Checking that we have a scratch dir set up
2025-09-14 07:04:07 ok 1 - Scratch dir is set (/tmp/9pm_i78hxc2f)
2025-09-14 07:04:07 ok 2 - Scratch directory actually exists
```

2.26. PASS db_write.tcl

Output

```
Starting: unit_tests/lib_tcl/db/db_write.tcl cmdl-supplied

2025-09-14 07:04:07 1..1
2025-09-14 07:04:07 # INFO - Writing to database
2025-09-14 07:04:07 ok 1 - Database written
```

2.27. PASS db_read.tcl

Output

```
Starting: unit_tests/lib_tcl/db/db_read.tcl cmdl-supplied

2025-09-14 07:04:07 1..2
2025-09-14 07:04:07 # INFO - Checking database content
2025-09-14 07:04:07 ok 1 - Database content: There is no spoon
2025-09-14 07:04:07 ok 2 - Database content: Take the red pill
```

2.28. ~~SKIP~~ OK Skip

Output

```
Starting: unit_tests/lib_tcl/tap/tap.tcl ok skip cmdl-supplied

2025-09-14 07:04:07 1..2
2025-09-14 07:04:07 ok 1 - This is a ok TAP test point
2025-09-14 07:04:07 ok 2 # skip This is a skip TAP test point
```

2.29. ~~SKIP~~ OK Skip OK

Output

```
Starting: unit_tests/lib_tcl/tap/tap.tcl ok skip ok cmdl-supplied

2025-09-14 07:04:07 1..3
2025-09-14 07:04:07 ok 1 - This is a ok TAP test point
2025-09-14 07:04:07 ok 2 # skip This is a skip TAP test point
2025-09-14 07:04:07 ok 3 - This is a ok TAP test point
```

2.30. ~~SKIP~~ Skip

Output

```
Starting: unit_tests/lib_tcl/tap/tap.tcl skip cmdl-supplied

2025-09-14 07:04:08 1..1
2025-09-14 07:04:08 ok 1 # skip This is a skip TAP test point
```

2.31. PASS OK

Output

```
Starting: unit_tests/lib_tcl/tap/tap.tcl ok cmdl-supplied

2025-09-14 07:04:08 1..1
2025-09-14 07:04:08 ok 1 - This is a ok TAP test point
```

2.32. PASS Plan First

Output

```
Starting: unit_tests/lib_tcl/tap/plan.sh first cmdl-supplied

2025-09-14 07:04:08 first cmdl-supplied
2025-09-14 07:04:08 0..1
2025-09-14 07:04:08 ok 1 - Supplying plan before test
```

2.33. PASS Plan Last

Output

```
Starting: unit_tests/lib_tcl/tap/plan.sh last cmdl-supplied

2025-09-14 07:04:08 last cmdl-supplied
2025-09-14 07:04:08 ok 1 - Supplying test before plan
2025-09-14 07:04:08 0..1
```

2.34. ~~FAIL~~ Plan Omit

Output

```
Starting: unit_tests/lib_tcl/tap/plan.sh omit cmdl-supplied

2025-09-14 07:04:08 omit cmdl-supplied
2025-09-14 07:04:08 ok 1 - Omitting plan
```