

TESSY V4.3 Features

New features in TESSY V4.3 (compared to TESSY V4.2)

Frank Büchner, October 2020 --- 001

Contents

1	Mutation Testing.....	2
2	Test Environment Editor (TEE)	4
3	Pattern for the Test Result	5
4	Tests without Instrumentation	5
5	The Author	5

1 Mutation Testing

TESSY automates mutation testing.

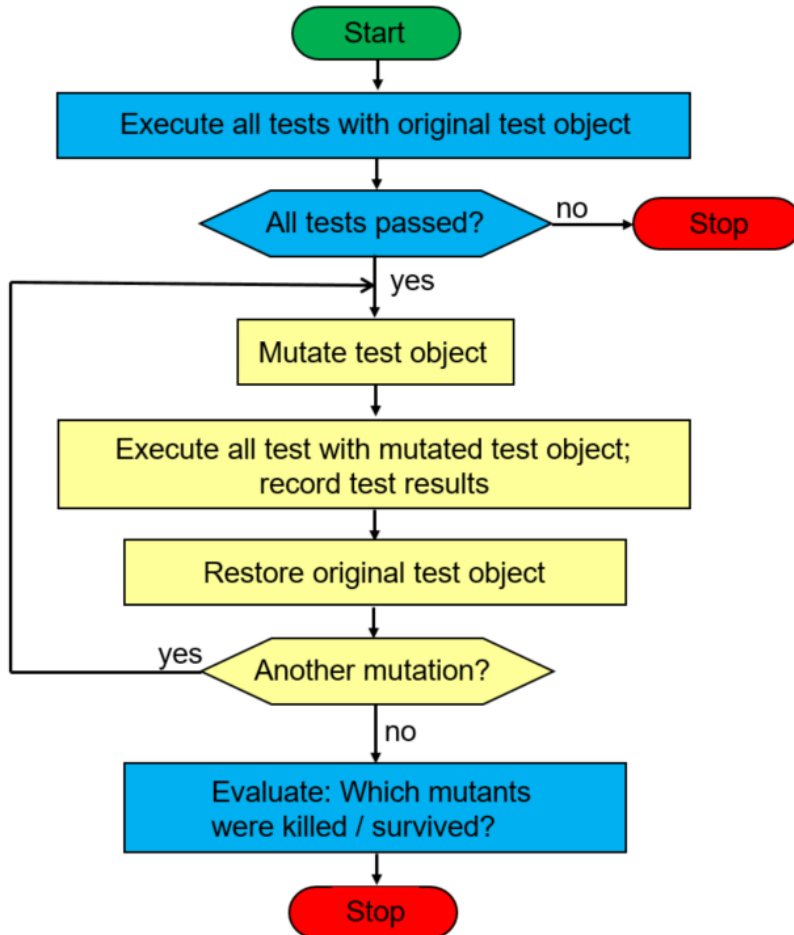


Fig. 1 The mutation test cycle

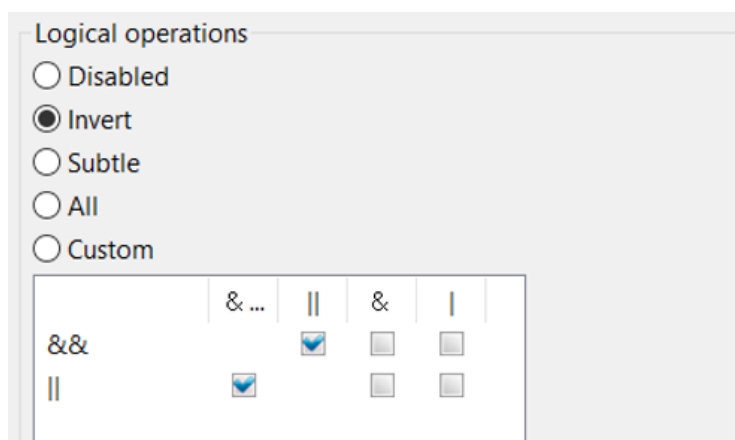


Fig. 2 The available mutations for logical operators

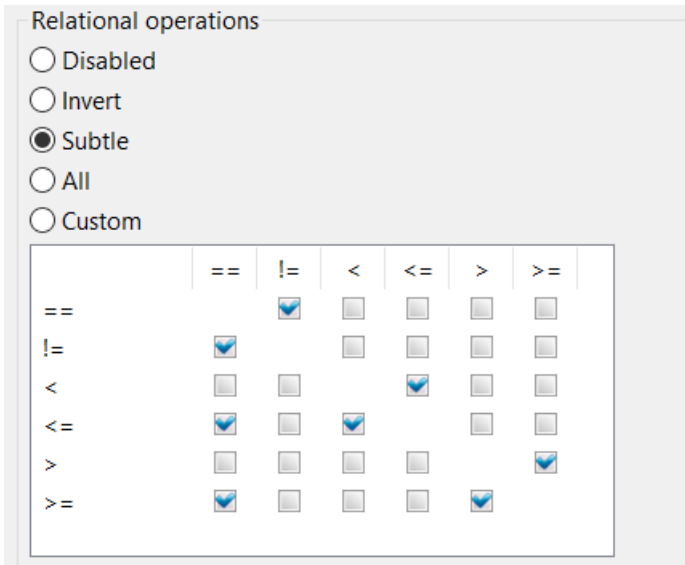


Fig. 3 The available mutations for relational operators

Mutation	Operation	Result
IF		✓
v1 < r1.range_start	<	✓
v1 <= r1.range_start	<=	✓
IF+IF		✗
v1 > (r1.range_start + r1.rang >	>	✗
v1 >= (r1.range_start + r1.>=	>=	✗

Mutation caused test failure

```

24 result is_value_in_range (struct range r1, int v1)
25 {
26   if (v1 < r1.range_start)
27     return no;
28
29   if (v1 > (r1.range_start + r1.range_len))
30     return no;
31
32   return yes;
33 }

```

Mutation	Operation	Result
IF		✓
v1 < r1.range_start	<	✓
v1 <= r1.range_start	<=	✓
IF+IF		✗
v1 > (r1.range_start + r1.rang >	>	✗
v1 >= (r1.range_start + r1.>=	>=	✗

Mutation survived all test cases

```

24 result is_value_in_range (struct range r1, int v1)
25 {
26   if (v1 < r1.range_start)
27     return no;
28
29   if (v1 > (r1.range_start + r1.range_len))
30     return no;
31
32   return yes;
33 }

```

Fig. 4 Mutation test result

2 Test Environment Editor (TEE)

The TEE is now a perspective in TESSY, and no longer a separate application.

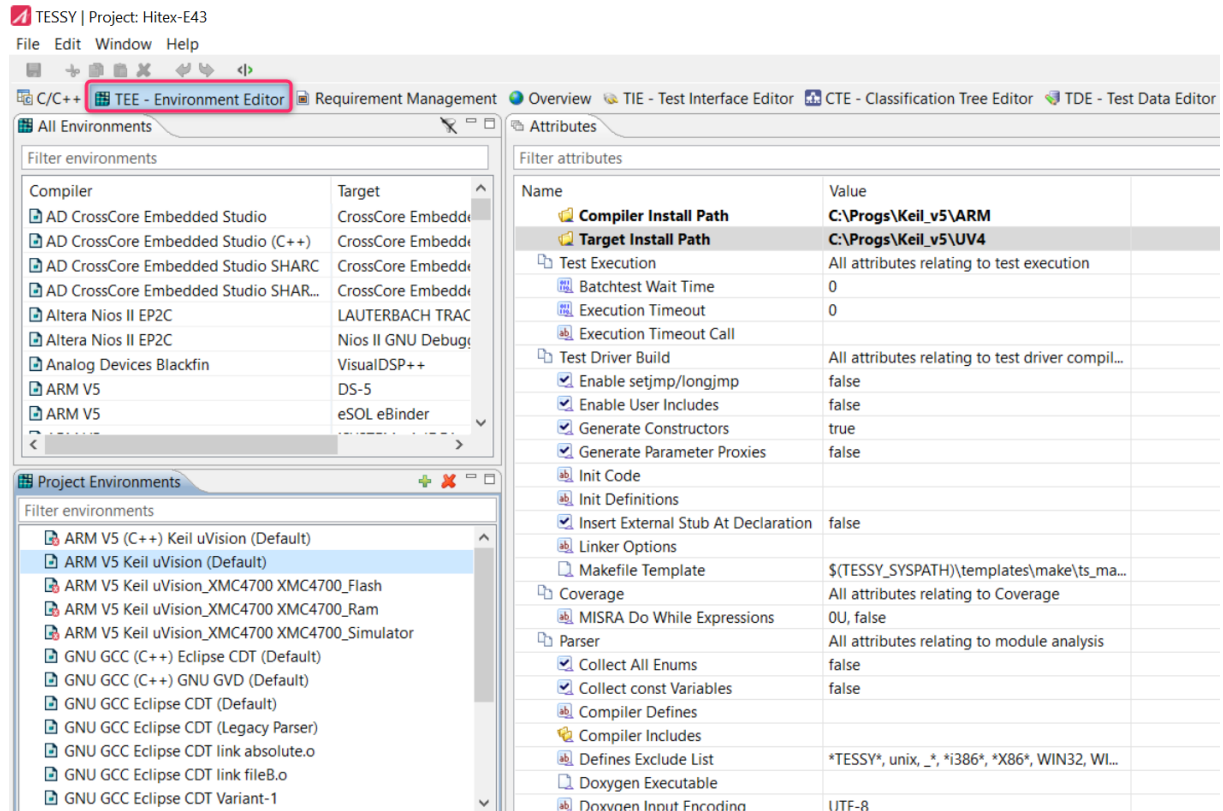


Fig. 5 TEE perspective in TESSY

3 Pattern for the Test Result

Prior to a test, the memory area of the test result is initialized with a pattern. The tests are repeated two times, with different patterns. This shall make sure that the test result is not passed by chance, because a passed test has definitely written the test result properly.

4 Tests without Instrumentation

Tests are run without instrumentation, to prevent tests that are passed because of instrumentation. This verifies that the instrumentation by TESSY was done correctly.

Instrumentation is needed for:

- Code Coverage
- Call Trace
- Fault injection
- Local static variables

This feature is especially useful for verifying the behavior of code instrumented for coverage vs. the original source code.

5 The Author

Frank Büchner, Hitex GmbH, frank.buechner@hitex.de



Any comments or questions to this document are welcome.