TESSY V4.0 Features

New features in TESSY V4.0 (compared to TESSY V3.2)

Frank Büchner, November 2016 --- 004

Contents

 © 2016 Hitex GmbH

Embedding Software Quality

1 New Technology for Testing of Software Written in C++

A substantial improvement for the testing of software written in C++ is one of the major features of TESSY V4.0. The completely new C/C++ code analyzer provides complete interface analysis of classes and their methods. Because of the complete interface analysis TESSY V4.0 can provide comfortable selections for constructors, what eases test case definition.

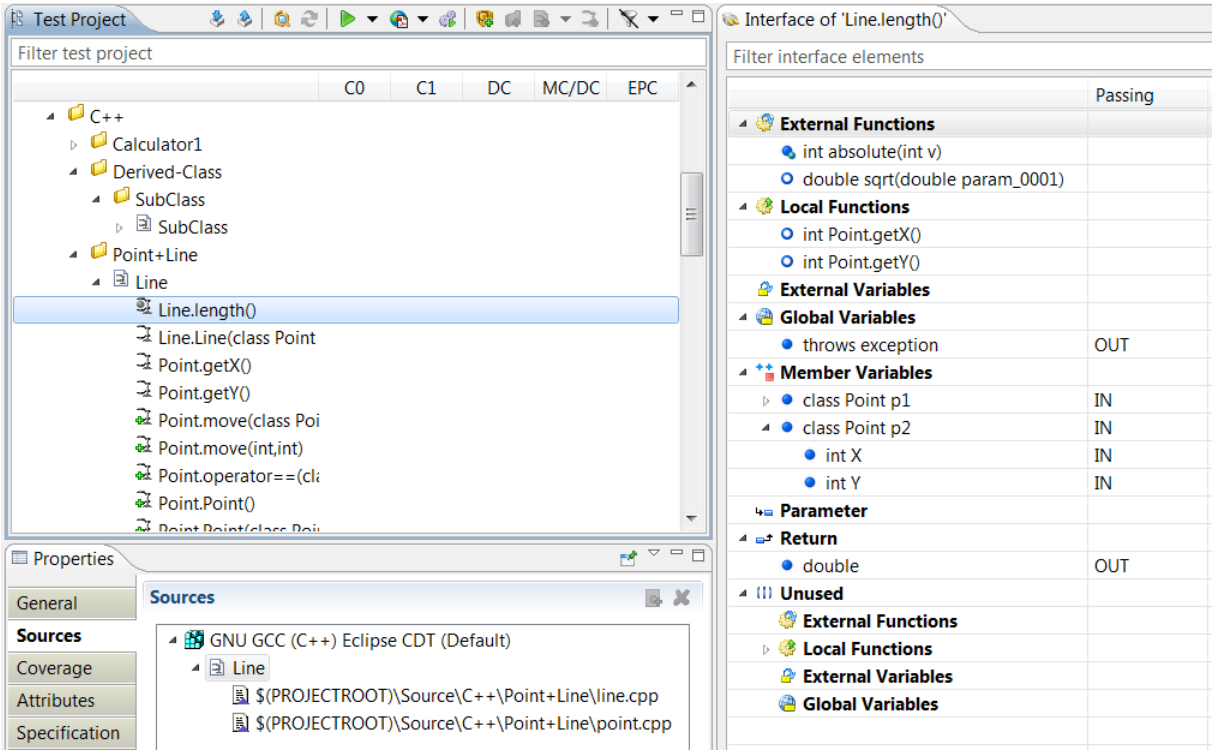

Fig. 1: The interface of a test object in C++ features C++ specific sections

TESSY supports stubbing of external and internal methods. Also template classes can be tested.

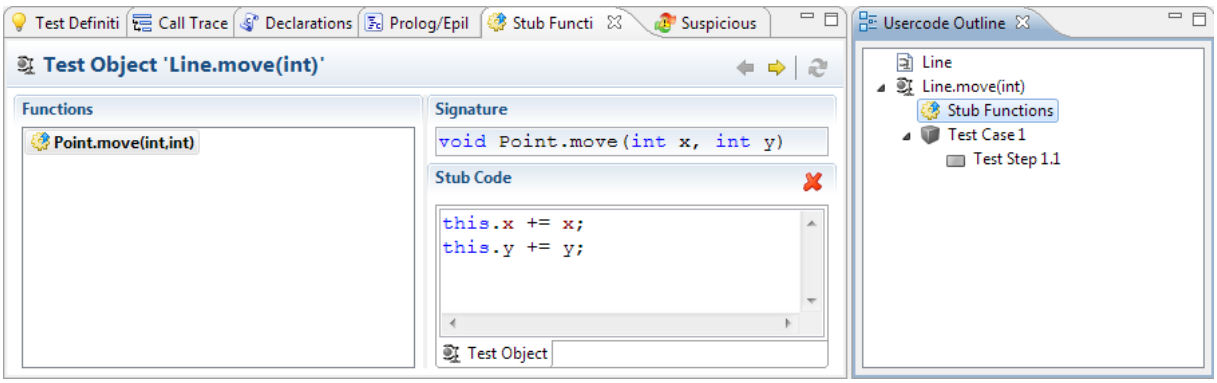

Fig. 2: Stub function for the method Point.move()

Embedding Software Quality

TESSY can create class instances using the constructors of the tested class in the Test Data view. There also parameter values of the tested method can be entered easily.

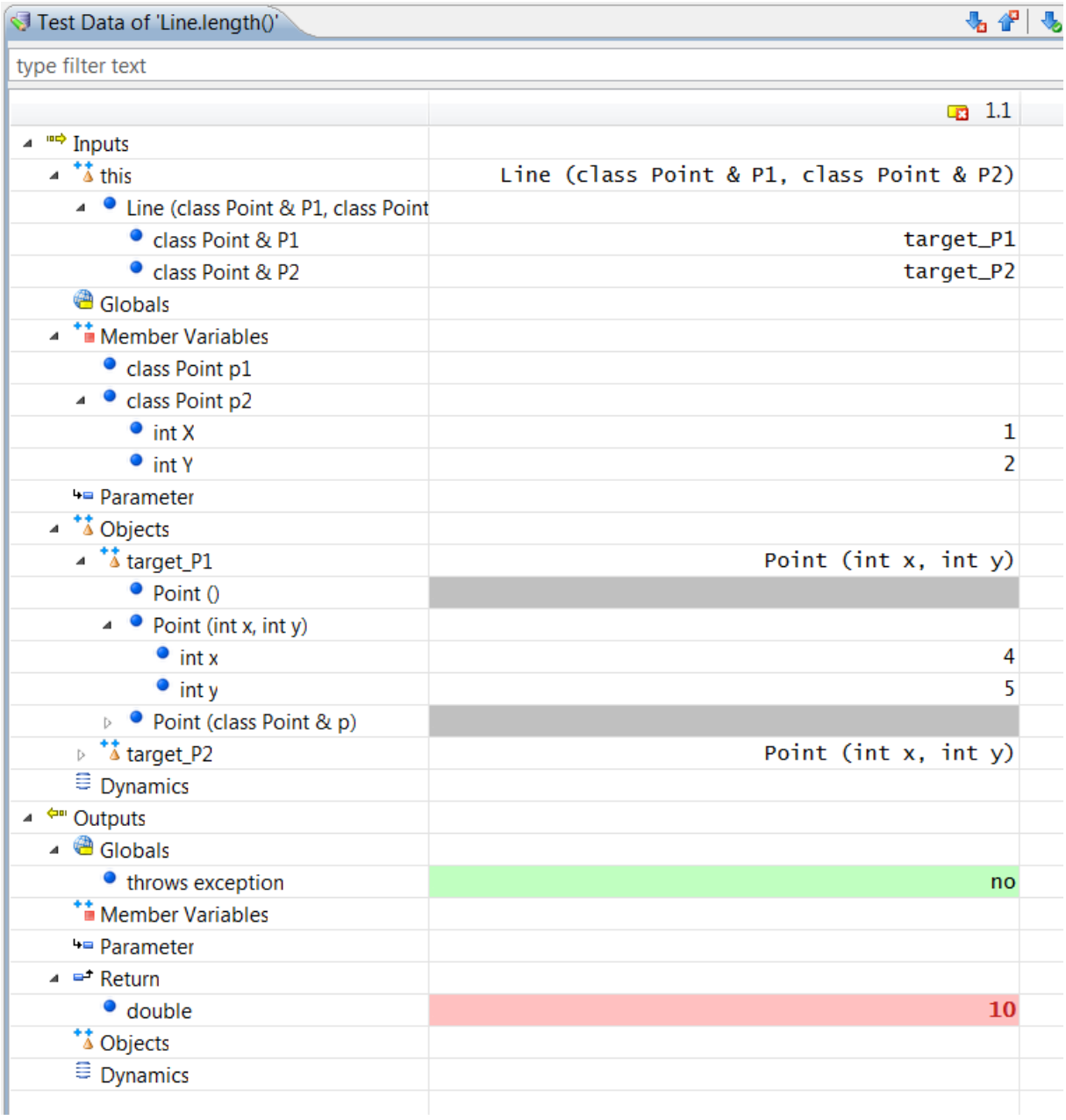

Fig. 3: A test case for the method Line.length()

 © 2016 Hitex GmbH

Embedding Software Quality

2 Management of Code Variants

Testing of code variants is now especially supported using derived modules according to a hierarchically structured variant tree.

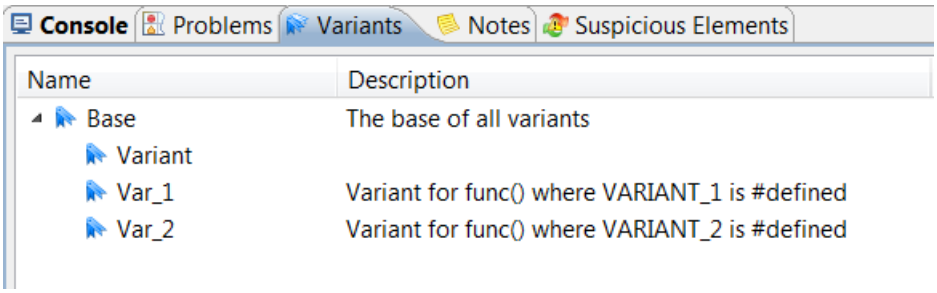

Fig. 4: The variant tree

A predefined root variant (i.e. the base of all variants) is automatically available for each project. Code variants can be defined as children of the base variant. Modules can be assigned to variants to indicate the relationship to a specific code variant to be tested.

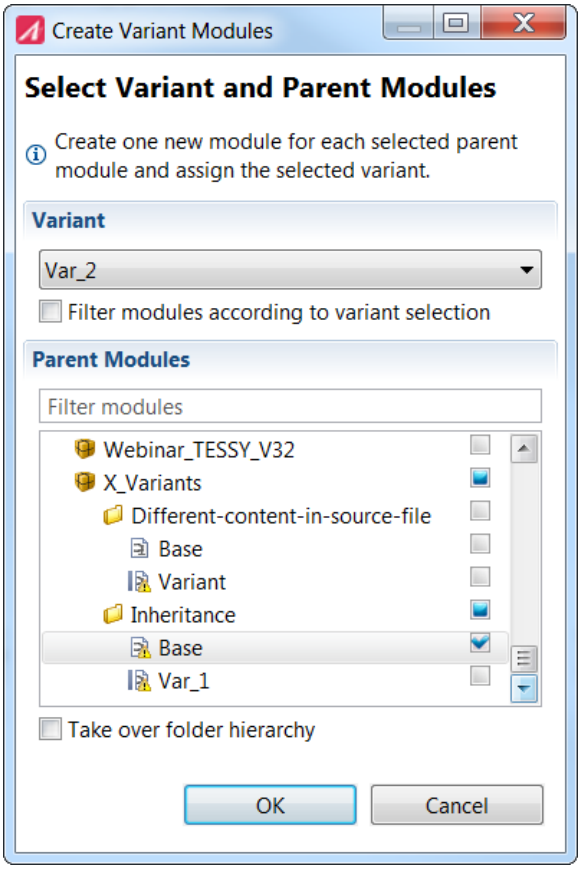

Fig. 5: Creating variant "Var_2" to the TESSY module "Base"

© 2016 Hitex GmbH

Embedding Software Quality

Each module that shall inherit test cases from a base module (i.e. a module assigned to a base variant) must also have a parent module assigned. This enables automated reuse of test cases defined within the parent modules. Basic test cases can be defined within base modules from which other variant modules can inherit.

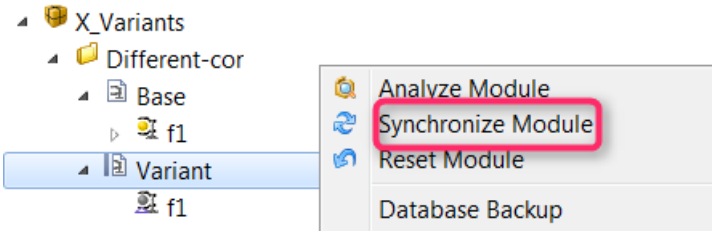

Fig. 6: Inherited test cases are synchronized with the test cases of the parent module

On interface changes or changes of the base test cases, all test data can easily be synchronized with the derived variant modules. Within the derived variant modules, test cases can be added, deleted and individual test data values can be overwritten (highlighted in purple as shown below).

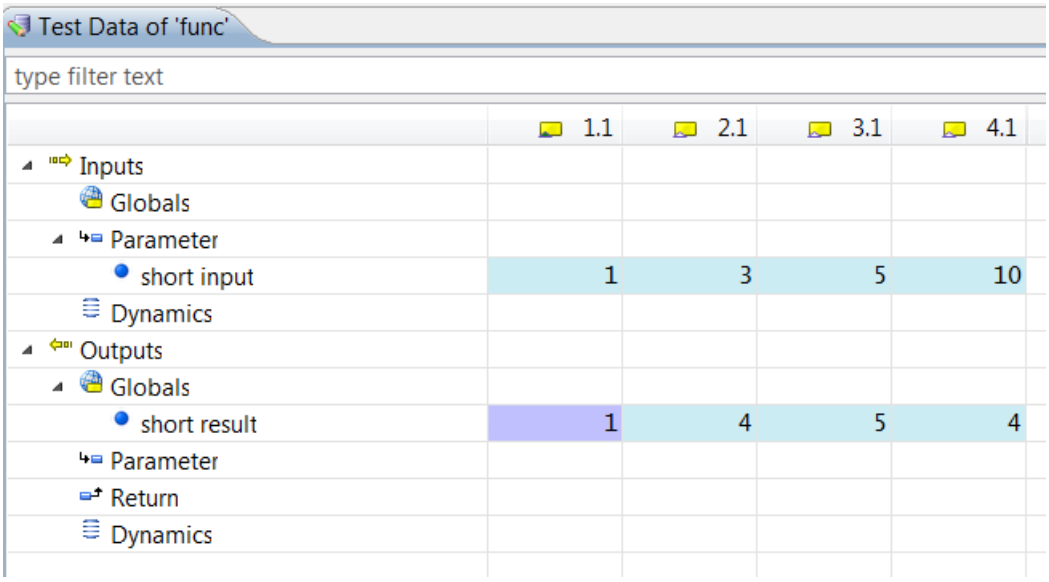

Fig. 7: The derived variant can have different values than those inherited

Embedding Software Quality

3 Test Objects without Source Files

To support Test-First Programming and Test-Driven Development (TDD) TESSY V4.0 lets you now create test objects without having the source file yet. Former TESSY versions required a source file and determined the test objects by analysis of this source file.

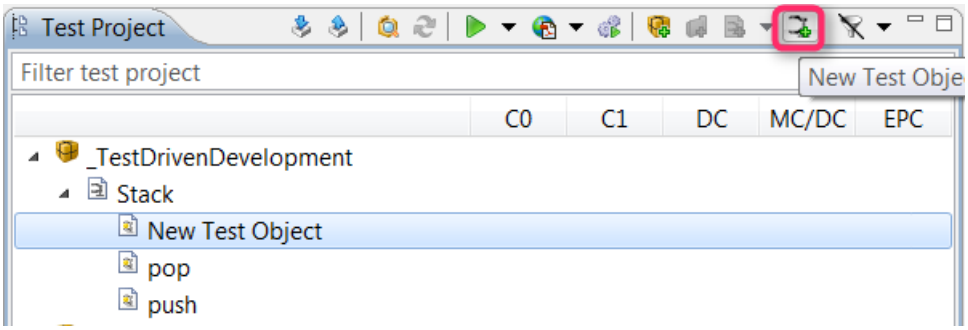

Fig. 8: Creating test objects without source file supports TDD

After having created the test object, you can add artificial variables to the interface of it. Then you can create test cases and specify test data for the artificial variables. Naturally, it is not possible to execute these tests because the source code of the test object still is not present.

After the source file with the actual test object is assigned to the TESSY module, you can assign the test cases for the artificial test object to the actual test object. This is even possible if the names of the artificial and the name of the actual test object differ and likewise if the names (and data types) of artificial and actual variables differ. These assignments are made by using the Interface Data Assigner (IDA) in TESSY.

4 Enhancements for Command Line Execution

Commands to set/unset/list compiler options and to set/unset comment, description, or specification (of test collection, folder, module, or test object) are added to the command line Interface, i.e. to the commands that are accepted by tessycmd.exe.

Furthermore, the behaviour of the command which exports test cases to an Excel workbook was changed in case the module does not have any test cases. Hitherto, a completely empty “Value” sheet was created; with TESSY V4.0 the interface of the test object is displayed in the “Values” sheet (by exporting a dummy test case). This allows creation of test cases through the command line interface (by creating them comfortably in the Excel “Values” sheet and then importing the Excel workbook back into TESSY).

© 2016 Hitex GmbH

5 UUID for TESSY Items

The following items of TESSY will get a Universally Unique Identifier (UUID) assigned to them:

- Test collections and folders
- Modules
- Test objects
- Test cases and test steps

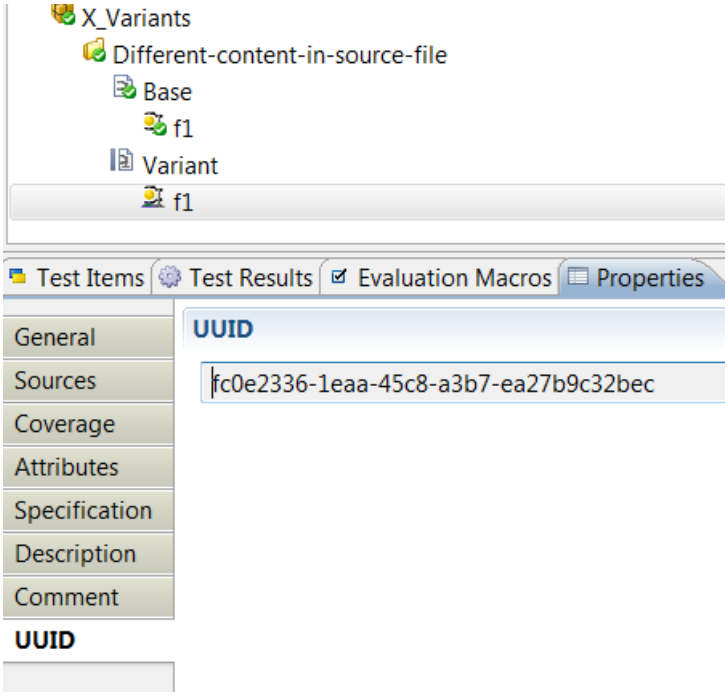

Fig. 9: The UUID of a test object

This allows identifying them even after save and restore via TMB files on different computers.

Existing projects will be updated automatically when opened using TESSY V4.0.

6 Optimized Test Execution

In order to optimize testing of large projects on continuous integration servers, the required memory resources during test execution have been minimized and the compilation of test drivers as well as the test run itself can now be performed in parallel. The latter pays off on computers with several kernels.

7 Auto-reuse During Non-interactive Execution

Enhanced auto-reuse in command line execution mode provides more stable regression test results in case of slightly changed interfaces as follows:

1. New unused struct or union members
2. New unused defines
3. Changed called function hierarchy
4. Removed interface elements or defines

8 Excluding Tests from Test Execution

Tests that shall not be executed for some reason (e.g. because they would fail due to a division by zero) can be excluded from the test execution. Such tests are displayed greyed out and they are automatically skipped when executing all tests of a test object.

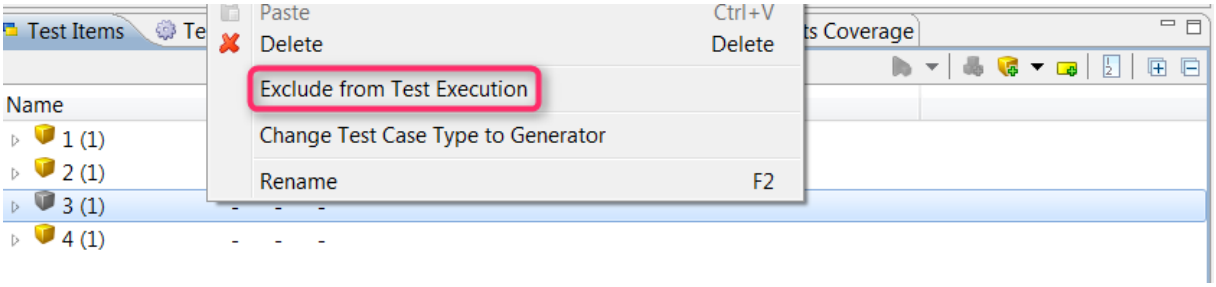

Fig. 10: Test case 3 is excluded from execution

9 More SVN Keywords Recognized

TESSY now recognizes more SVN keyword in the source file than in TESSY V3.2. Hitherto, only Author and Revision was recognized. TESSY V4.0 recognizes additionally keywords and displays them in the test report.

Source Code	Excerpt of Test Report

`// $Revision: 3.0 $`
`// $Author: Br $`
`// $Id: is_value_in_range.c  variant A$`
`// $Date: 2016-10-30 $`
	Revision: 3.0 Author: Br Date: 2016-10-30
	is_value_in_range.c variant A

Fig. 11: The values of the keywords of the source code (left hand side) are included in the test report (right hand side)

10 Highlight Modified Variables

In the Test Data view of the TDE perspective, a new button now allows you to keep track of the changes that you made with respect to test data.

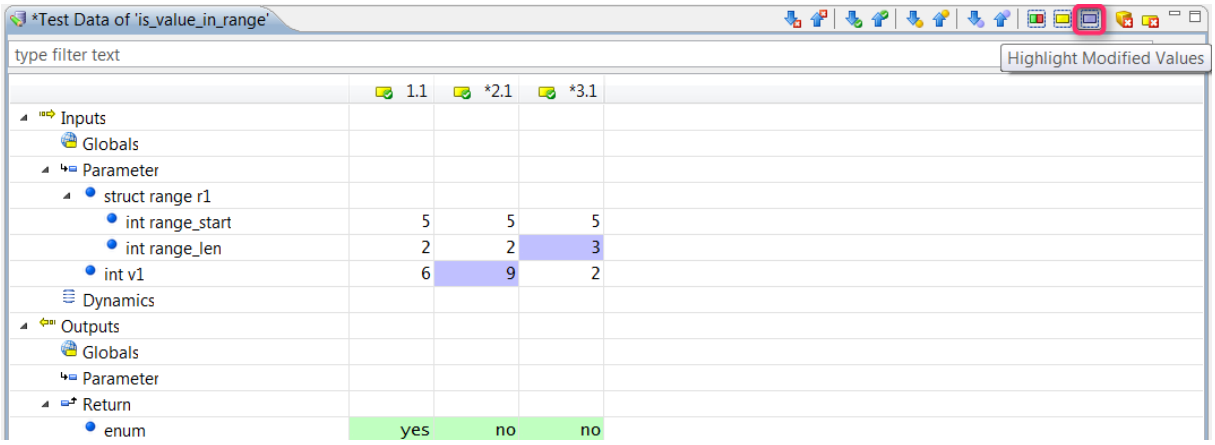

Fig. 12: In test case 2.1 the value of v1 was changed to 9; in test case 3.1 the value of range_len was changed to 3

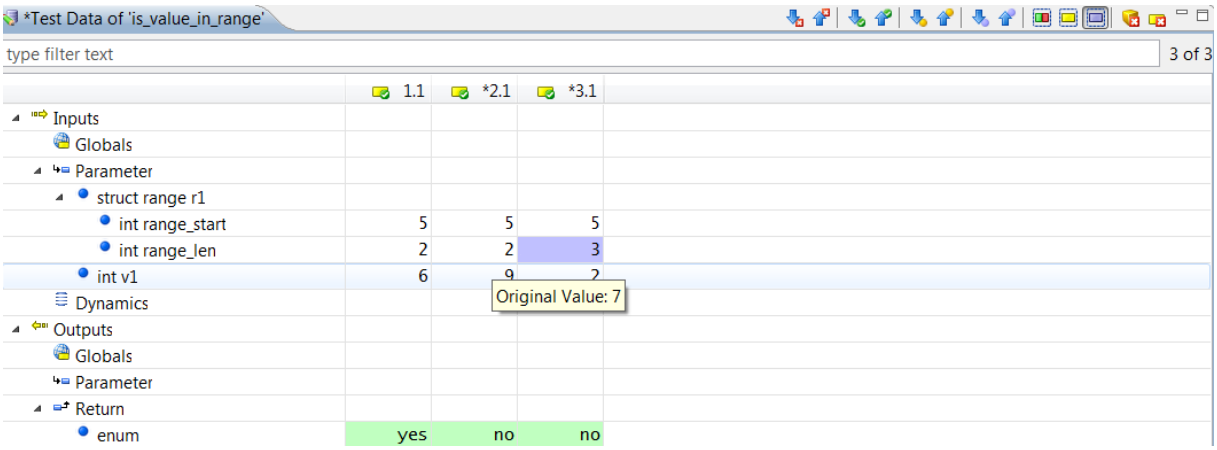

Fig. 13: In test case 2.1 the original value of v1 was 7

 © 2016 Hitex GmbH

Embedding Software Quality

11 Add Include Paths

Several include paths can be selected at the same time and are added by one mouse click to the list of directories where TESSY searches for header files

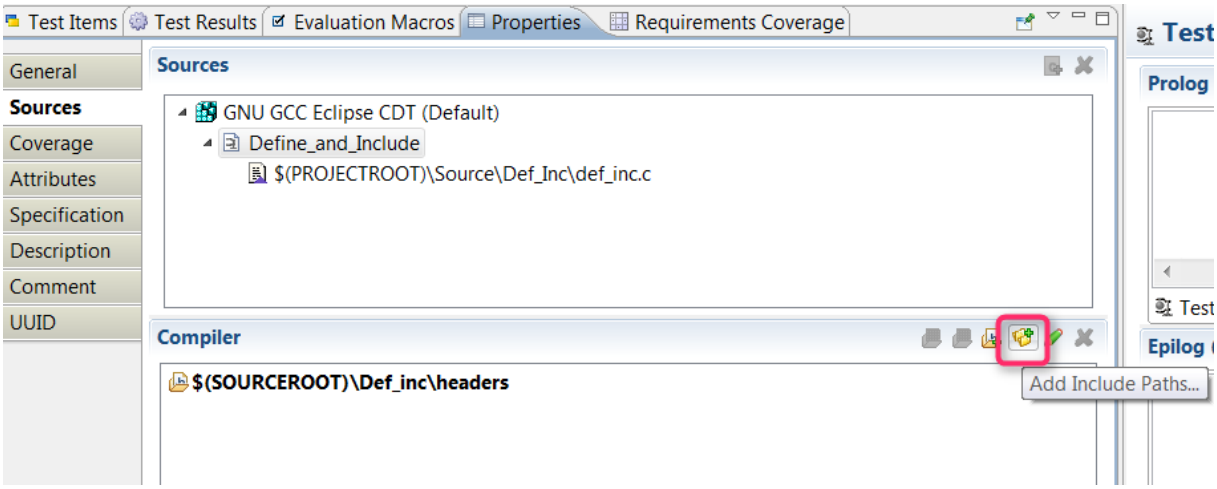

Fig. 14: Add several include paths at the same time

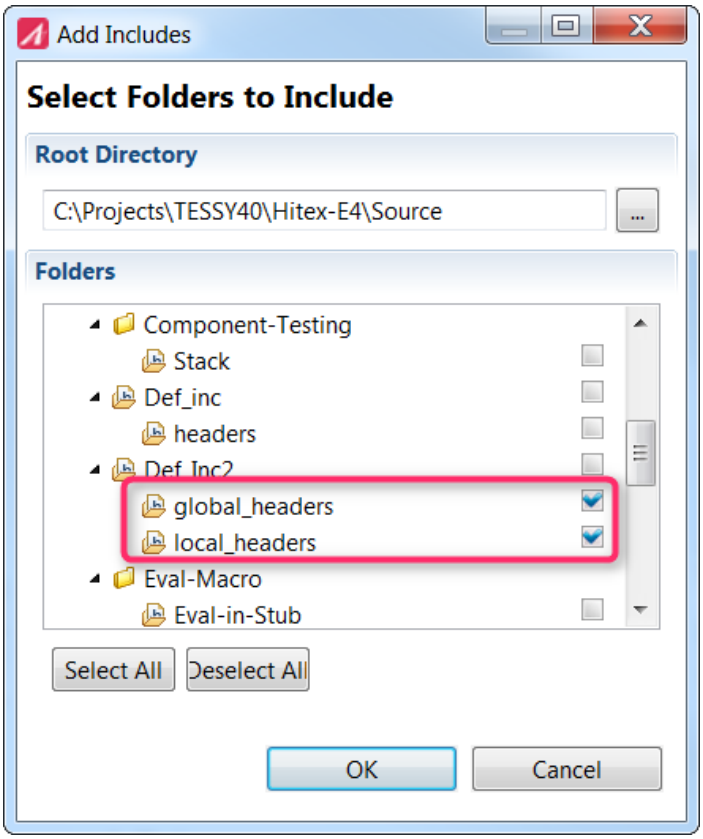

Fig. 15: Select folders to include

© 2016 Hitex GmbH

Embedding Software Quality

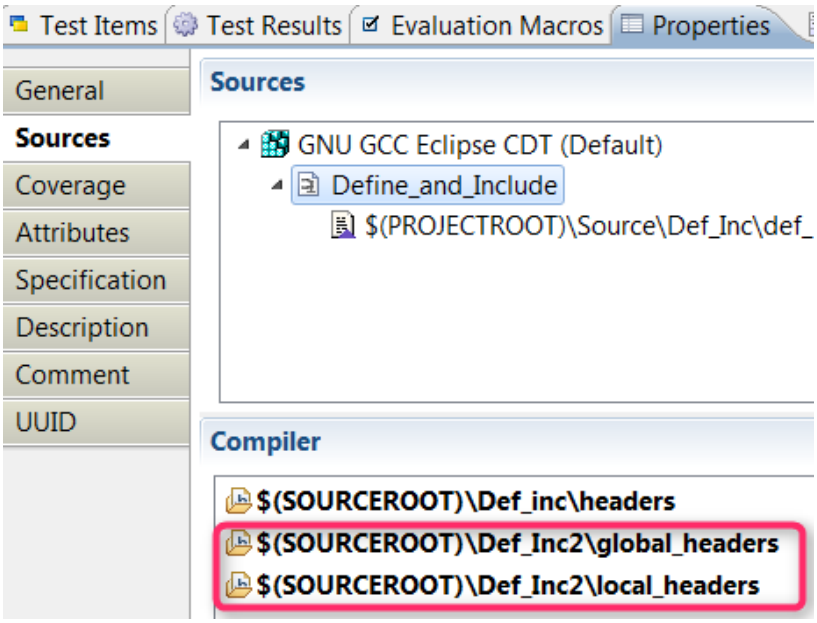

Fig. 16: Two directories were added at the same time

12 The Author

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

Any comments or questions to this document are welcome.

Embedding Software Quality