

Tessy V2.9 Features

Please note: The version number V2.7 and V2.8 were never used.

Contents

1	Temporal Component Testing / Integration Testing	1
2	Accelerated Interface Analysis	3
3	Optimized Test Database	3
4	Comfortable Save/Restore	3
5	Improved Support for C99	4
6	Enumerations as Base Type for Bitfields.....	4
7	Coverage Measurement Is Improved	5
8	Command Line Import of Test Data.....	5

1 Temporal Component Testing / Integration Testing

Component testing will allow testing interacting units. Component testing is integration testing for the units in the component.

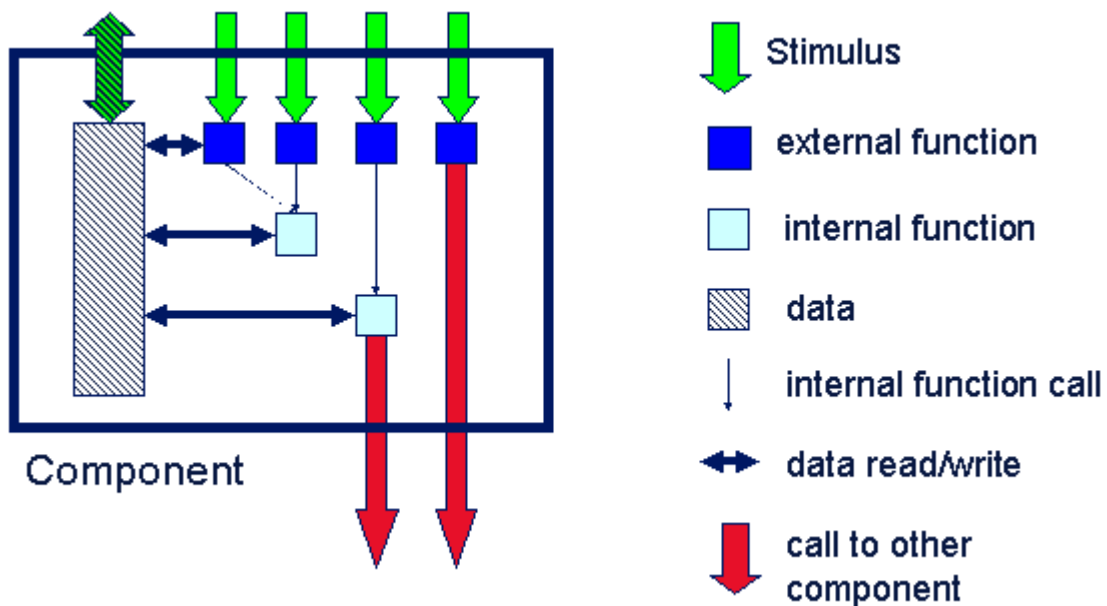


Fig. 1 Scheme of a component

A test case for a component is called “scenario”. It consists of a sequence of stimulating calls to the component, intermixed with resulting calls from the component under test to other components. Also the data of the component that is accessible

from outside the component can be set in a scenario resp. evaluated for the test result in a scenario.

Also a scenario allows specifying the temporal behavior of the component.

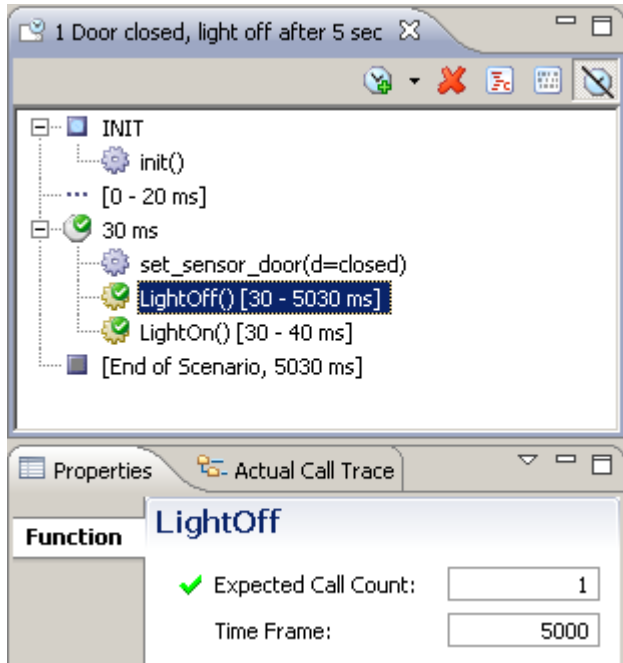


Fig. 2 Temporal behavior can be specified

Testcase 2 : Normal Operation

Teststep 2.1					
Name	Input Value				
number	0				
Name	Actual Value	Eval	Expected Value	Result	
number	0	==	0	✓	

Call Trace					
Actual Function	Count	Expected Function	Count	Result	
set_number [10 ms]	1	set_number [10 ms]	1	✓	
crossed_75 [40 ms]	1	crossed_75 [40 ms]	1	✓	
crossed_50 [60 ms]	1	crossed_50 [40-80 ms]	1	✓	
crossed_25 [70 ms]	1	crossed_25 [40-100 ms]	1	✓	

Fig. 3 Actual function calls are evaluated against expected time frames

2 Accelerated Interface Analysis

An improvement of the internal interface database results in higher speed of the interface analysis and faster access to the interface information.

3 Optimized Test Database

The internal database for test data has been redesigned for optimized performance.

4 Comfortable Save/Restore

The backup file(s) in TMB format can now be created for a complete database, for a complete project, or for a complete module in a single action. This eases a lot version control and the transfer of test data.

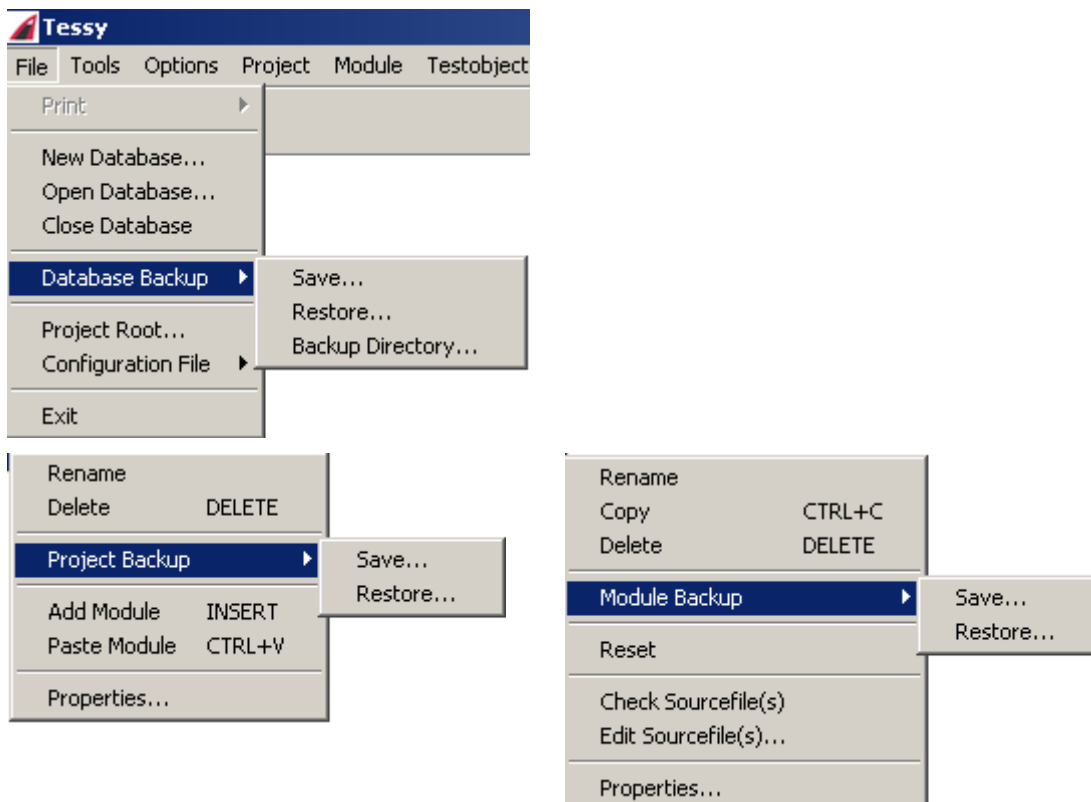


Fig. 4 New menus can be used to save the entire database (top left), an entire project (bottom left) or an entire module (bottom right)

5 Improved Support for C99

Designators in initialisers and a mix of declarations and statements in compound statements, both allowed by the C99 standard, are now supported by Tessy.

<pre>int a[] = {[0] = 0, [1] = 2}; int func1(void) { return(a[0]); }</pre>	<pre>{ int i; // definition i = 1; // statement int j; // definition j = 2; // statement . . . }</pre>
---	--

Fig. 5 Designators in initialisers (left hand side) and a mix of definitions and statements (right hand side) is now supported by Tessy

6 Enumerations as Base Type for Bitfields

If enumerations types are used as base type for bitfields, the enumerated names can now be used in the Test Data Editor (instead of the numeric values, as in former Tessy versions).

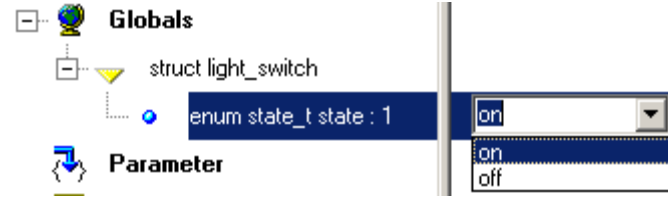
<pre>enum state_t { on, off }; struct { enum state_t state: 1; } light_switch;</pre>	
--	--

Fig. 6 Now identifiers from the enumeration can be used in the TDE (right hand side)

7 Coverage Measurement Is Improved

The rules for coverage measurement are tightened with respect to

1. the question mark operator (“?:”) and
2. conditions (Boolean expressions) in statements.

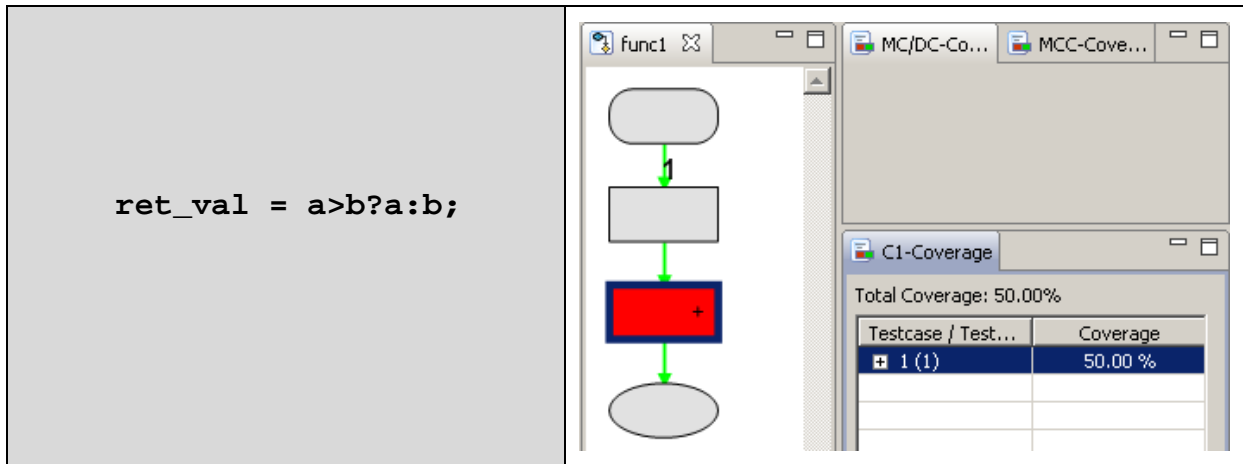


Fig. 7 Two test cases are needed in Tessy V2.9 in order to get 100% branch coverage for the question mark operator

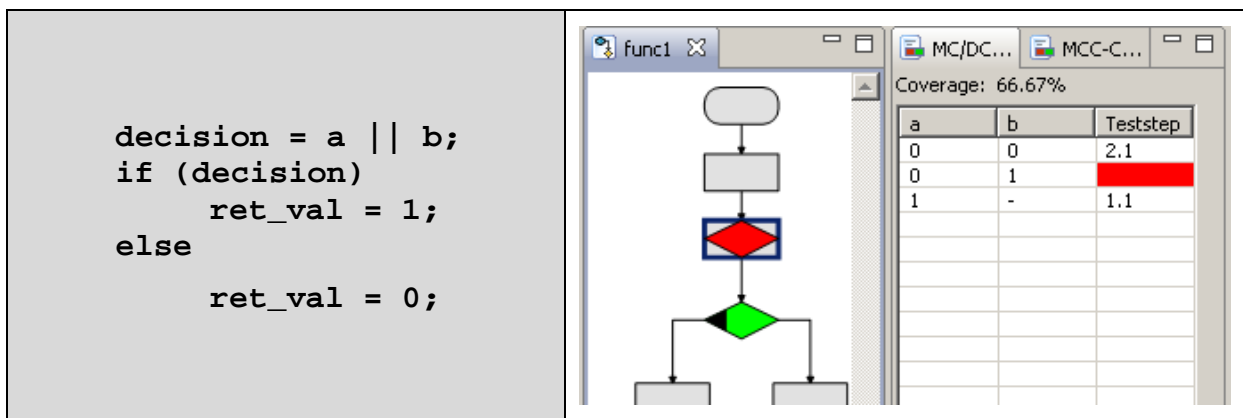


Fig. 8 Three test cases are needed in Tessy V2.9 in order to get 100% MC/DC

8 Command Line Import of Test Data

Since Tessy V2.9.20, it is possible to import test data using the Tessy command line tool tdbcmd. This allows the non-interactive import of test data from Excel workbooks or text files (ASCII). The non-interactive functionality is comparable to the interactive functionality. This feature eases the operation of Tessy by another tool.