



Integration Between BenderRBT and TestExplorer

TestExplorer is a test management tool aimed at manual testing. More information can be found about the product and the company at <http://www.Sirius-SQA.com>.

Most software testing tools can be divided into the following seven activities:

- ◆ **Define Test Completion Criteria:** The test effort has specific, quantifiable goals. Testing is completed only when the goals have been reached (e.g., testing is complete when the tests that address 100% functional coverage of the system all have executed successfully).
- ◆ **Design Test Cases:** Logical test cases are defined by four characteristics: the initial state of the system prior to executing the test; the data; the inputs; and the expected results.
- ◆ **Build Test Cases:** There are two parts needed to build test cases from logical test cases: creating the necessary data; and building the components to support testing (e.g., build the navigation to get to the portion of the program being tested).
- ◆ **Execute Tests:** Execute the test case steps against the system being tested and document the results.
- ◆ **Verify Test Results:** Testers are responsible for verifying two different types of test results: are results as expected; and do the test cases meet the test completion criteria.
- ◆ **Verify Test Coverage:** Track the amount of functional coverage achieved by the successful execution of each test.
- ◆ **Manage the Test Library:** The test manager maintains the relationships between the test cases and the programs being tested. The test manager keeps track of what tests have/have not been executed, and whether the executed tests have passed or failed.

Integration Between BenderRBT and TestExplorer

Table 1 describes which of these testing activities are addressed by BenderRBT and which are addressed by Sirius-SQA's Test Explorer.








Test Activity	BenderRBT	Test Explorer
Define Test Completion Criteria	BENDER 	
Design Test Cases	BENDER 	
Build Tests		 TESTEXPLORER
Execute Tests		 TESTEXPLORER
Verify Test Results		 TESTEXPLORER
Verify Test Coverage	BENDER 	
Manage Test Library		 TESTEXPLORER

Table 1: Test Activities

BenderRBT designs the minimum number of test cases that provide 100% functional coverage for the system under test. The benefits of BenderRBT and its integration with TestExplorer are described below:

1. Requirements-Based Testing stabilizes the functional definition and therefore stabilizes the user interface earlier in the software development process. This means you can invest in implementing your scripts without fear of major scrap and rework occurring from requirements issues being uncovered late in a project.
2. BenderRBT minimizes the number of test cases to execute - twice the functional coverage with half the number of test cases compared to manual approaches to designing test cases. And, since it takes 3 to 5 times as long to build an executable

Integration Between BenderRBT and TestExplorer

test case than it does to design it, BenderRBT provides major resource and time savings in scripting tests.

Test Cases can be exported directly from BenderRBT to TestExplorer. This eliminates the time and effort to manually enter them into TestExplorer.

TestExplorer projects are maintained on the local desktop, or, in team environments, on a network share. A configuration file on the desktop maintains the location of the projects. The integration, a COM server, validates that TestExplorer is installed on the local machine, and then accesses this configuration file in order to find the TestExplorer projects. Once the RBT user selects a project for export and begins the export process, the integration creates fully formed TestExplorer tests and places them into the project, maintaining referential integrity in the TestExplorer system.

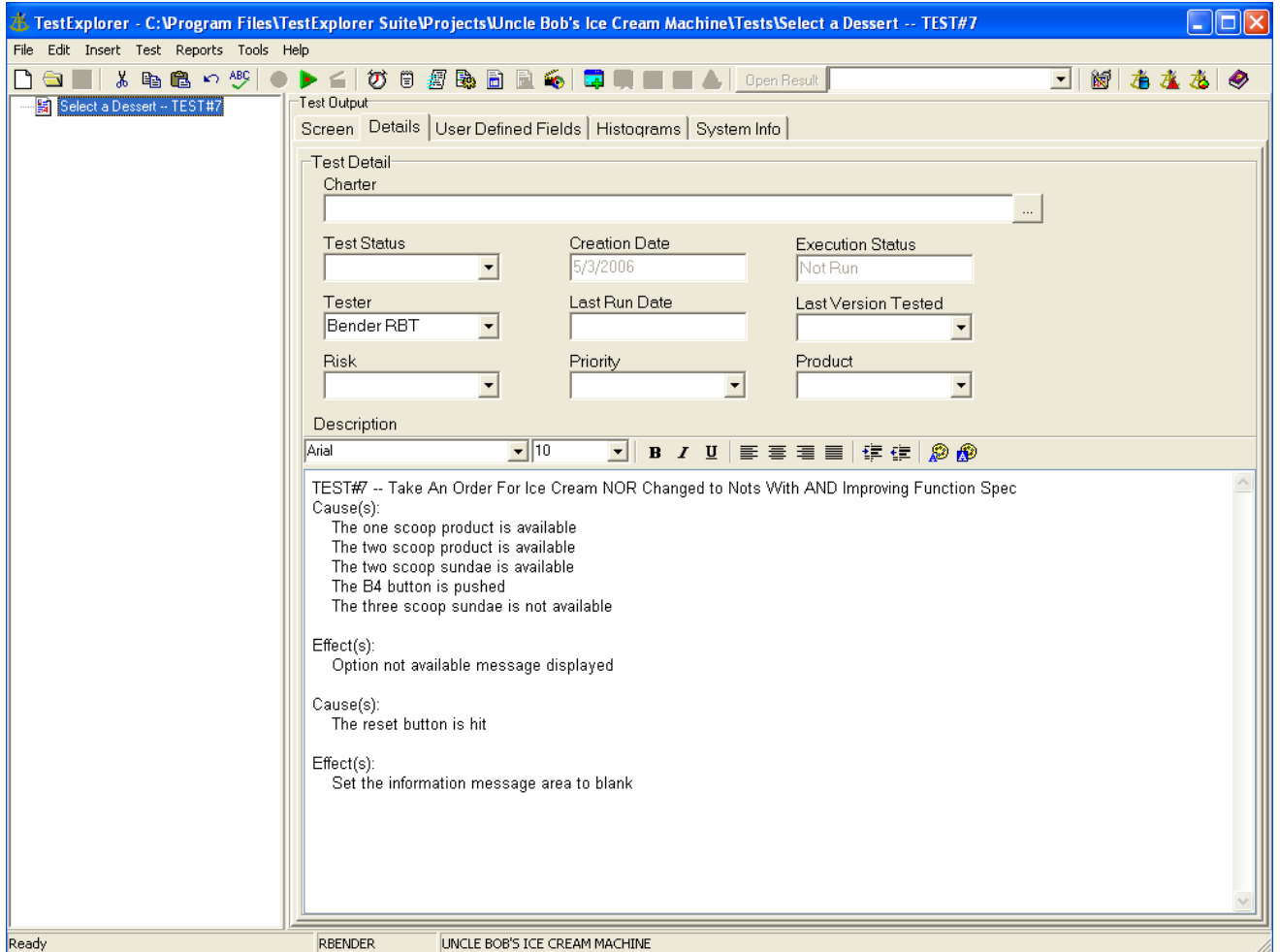
In BenderRBT selecting Utilities → Export to TestExplorer brings up the following dialog:



Export to TestExplorer Dialog

Once the tests have been exported, you can now find them in TestExplorer. Opening up the project and selecting one of the tests will show:

Integration Between BenderRBT and TestExplorer



Tests in TestExplorer

Maintaining the Test Library

When the application's rules change you update the models in BenderRBT. It then can tell you what changes need to be made to your existing test cases to bring them in sync with the new rules. This includes what modifications are required to existing tests and what new tests must be added to get back to full functional coverage. The key is that your investment in your test libraries is protected across releases as the system evolves. You never have to start over for each release of your application.

Integration Between BenderRBT and TestExplorer

BenderRBT provides the following reports to supplement TestExplorer:

- The **Script Test Case Definitions** report lists causes and effects in their logically sequence (i.e., they appear in groups of related causes followed by their associated effects for each test step of the test cases generated).
 - The **Batch Test Case Definitions** report first lists all of the causes, which are then followed by all of the effects for each of the test cases generated.
 - The **Coverage Matrix** report shows the functional variation coverage achieved by each test case.
 - The **Definition Matrix** report shows the causes and effects that make up each test case.
 - The **Statistics** report shows the computations of the percentage of feasible versus testable variation coverage achieved by the test cases and other demographic information.
 - The **Logic Diagram** report shows the Cause-Effect Graph, which shows the causes, effects, relations and constraints that describe the system for which the test cases are designed.
 - The **Functional Specification** report that restates the test cases in structured English
3. Since BenderRBT produces the appropriate set of test cases to cover 100% of the functionality for the system under test, once each of these test cases is successfully executed, then test execution is complete. BenderRBT also provides the ability to show how much functional coverage the execution of each test case contributes to the entire test effort.

With the Coverage Matrix report displayed, from the Utilities Menu, select the Coverage Analysis... Option. As test cases are selected (highlighted), the proportion of weak and strong functional coverage are calculated for that set of test cases.

Weak coverage denotes the simple percentage of ANY Functional Variations which have been covered by the selected (or completed) test cases.

Strong Coverage is tallied only for those Functional Variations where ALL of the functional variations derived from any given Relations statement are covered.

You can also use this feature to determine an optimal subset of tests to create and run. For example, you might not have time before a critical checkpoint to build all of the tests. Which ones should you create first? BenderRBT can tell you this via the Fewer Tests option.

Integration Between BenderRBT and TestExplorer

V A R I A T I O N	T	T	T	T	T	T	T
	E	E	E	E	E	E	E
	S	S	S	S	S	S	S
	1	2	3	4	5	6	7
1	#						
2		X	X	X	X	X	X
3	Infeasible						
4		#					
5	X			X	X	X	X
6			#				
7				#			
8	X	X				X	X
9					#		
10							#
11	X	X		X			
12							#
13			X		X		X
14		#					
15		#					
16				#			
17							#
18			X				X
19	X	X		X		X	
20					#		
Unique Vars	2	2	1	2	2	2	1
Total Vars	6	6	4	6	5	6	6

Coverage Analysis

Weak Coverage:
12 / 19 * 100 = 63%

Strong Coverage:
3 / 19 * 100 = 15%

Note: Select = Any ONE Test Name
SHIFT+Select = RANGE of Test Names
CTRL+Select = MULTIPLE Test Names

[Fewer Tests >>](#)

Coverage Analysis Dialog

Integration Between BenderRBT and TestExplorer

V A R I A T I O N	T E S T #	T E S T #	T E S T #
	2	3	5
1			
2	X	X	X
3			
4	#		
5			X
6		#	
7			
8	X		
9			#
10			
11	X		
12			
13		X	X
14			
15	#		
16			
17			
18			X
19	X		
20			#
Unique Vars	2	2	1
Total Vars	6	6	4

Coverage Analysis

Weak Coverage:

Strong Coverage:

Note: Select = Any ONE Test Name
 SHIFT+Select = RANGE of Test Names
 CTRL+Select = MULTIPLE Test Names

Fewer Tests
 Number of Tests:

% Strong Coverage:

% Weak Coverage:

Maximize
 Strong
 Weak

100% Complete

Fewer Tests Dialog

The net is that BenderRBT and TestExplorer are totally complementary, providing their users with a powerful test management solution.