Integration Between BenderRBT and Mercury Tools

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.

Table 1 describes which of these testing activities are addressed by BenderRBT and which are addressed by Mercury Interactive's automated testing tools.

Table 1: Test Activities

Test Activity	BenderRBT	Mercury
Define Test Completion Criteria	BENDER RBT	
Design Test Cases	BENDER RBT	
Build Tests		MERCURY INTERACTIVE
Execute Tests		MERCURY INTERACTIVE
Verify Test Results		MERCURY INTERACTIVE
Verify Test Coverage	BENDER RBT	
Manage Test Library		MERCURY INTERACTIVE

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 TestDirector 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 WinRunner 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

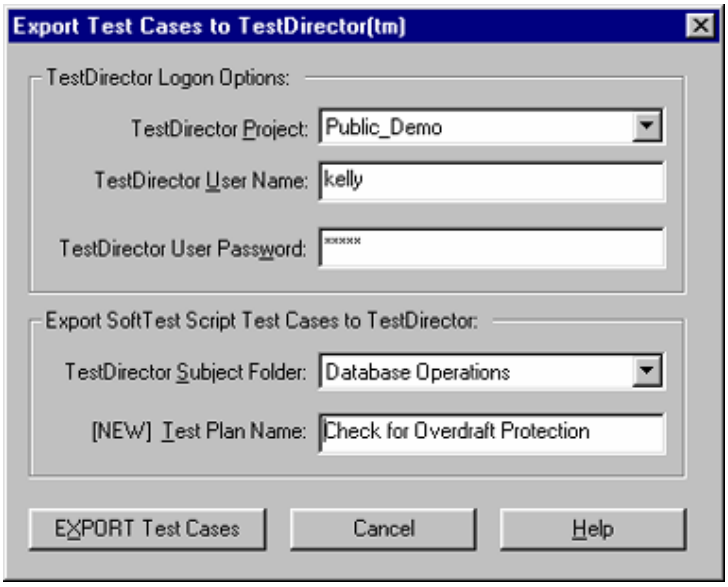
designing test cases. And, since it takes 3 to 5 times as long to script a test case than it does to design it, BenderRBT provides major resource and time savings in scripting tests.

3. Test Cases can be exported directly from BenderRBT to TestDirector. This eliminates the time and effort to manually enter the test steps into TestDirector. The test case export approach can be summarized as follows:

   - Create a TestDirector Project using the Project Administrator.
   - Open the project just created using TestDirector and create a TestDirector Subject Folder.
   - Design a set of test cases using BenderRBT.
   - Export the test cases from BenderRBT to TestDirector.
   - View the BenderRBT test cases exported to TestDirector.

Export Test Cases from BenderRBT to TestDirector

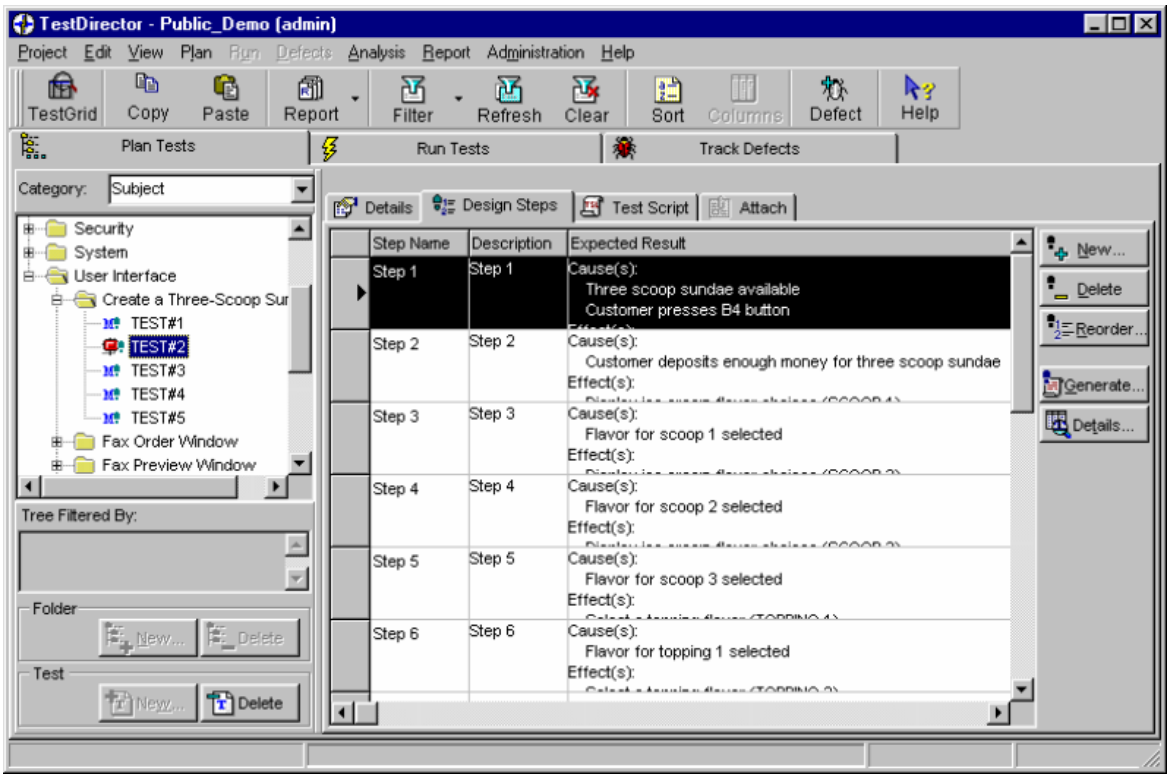
1) From BenderRBT, click the Utilities Menu and select the Export to TestDirector… Option.
2) From the Export Test Cases to TestDirector dialog box, select the appropriate TestDirector Project name.
3) Enter the TestDirector User Name.
4) Enter the TestDirector User Password.
5) Click the Connect to TestDirector Button. The Connect to TestDirector Button disables signifying that a connection has been made to TestDirector.
6) Select the appropriate TestDirector Subject Folder.
7) Enter a [New] Test Plan Name. The Export Test Cases Button enables.



8) Click the Export Test Cases Button. A BenderRBT dialog box displays a message signifying that the export has been completed.
9) Click the OK Button to close the BenderRBT dialog box.

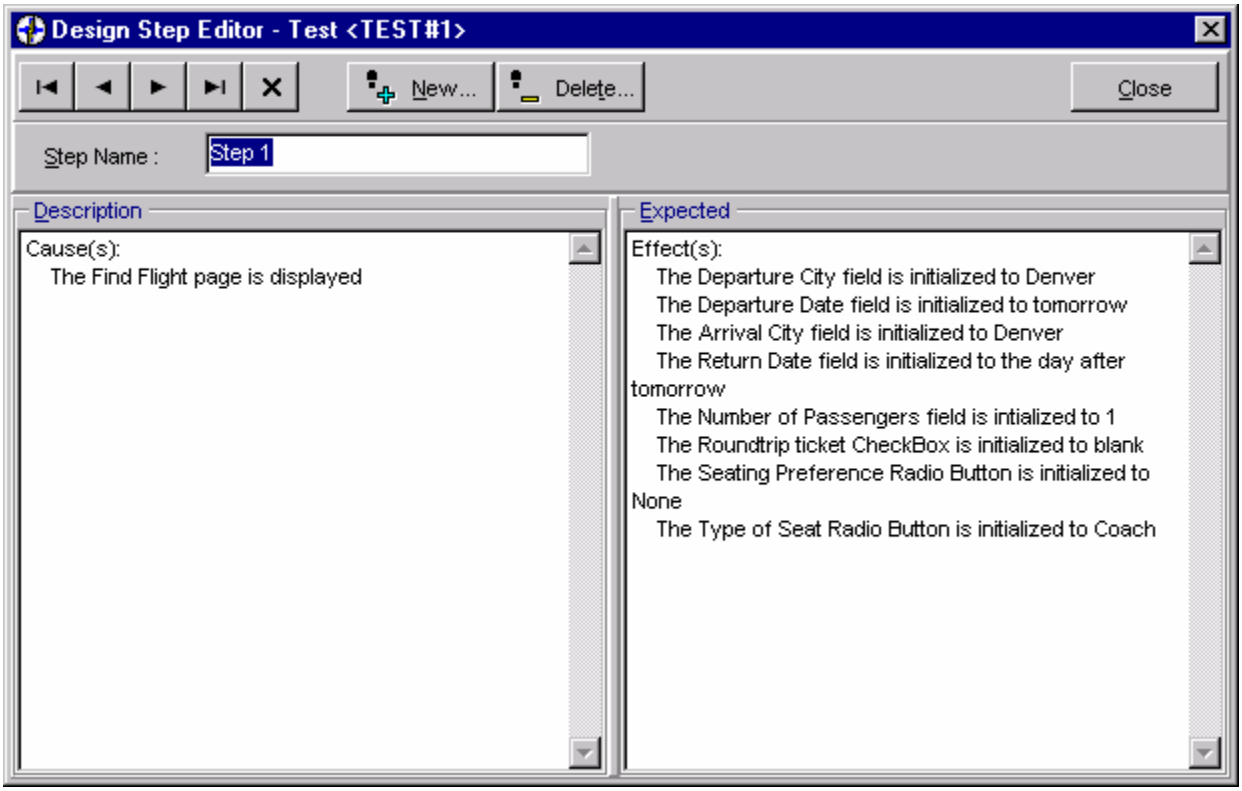
View BenderRBT Test Cases Exported to TestDirector.

1) If TestDirector is not already open, open it with the appropriate Project name. If TestDirector is already open, then click the Refresh Button.
2) Click on the appropriate folder.
3) The test plan and test cases exported from BenderRBT appear under the folder selected.



4) If a TEST is selected, and the Design Steps Tab is clicked, then the following test case information imported from BenderRBT is displayed.

Each test step in BenderRBT receives a Step number in the Step Name field in TestDirector. The causes in BenderRBT are exported to the Description field in TestDirector. The effects in BenderRBT are exported to the Expected Result field in TestDirector. If the Details… Button is clicked, the following test case information exported from BenderRBT is displayed:



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.

BenderRBT provides the following reports to supplement TestDirector:

- 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

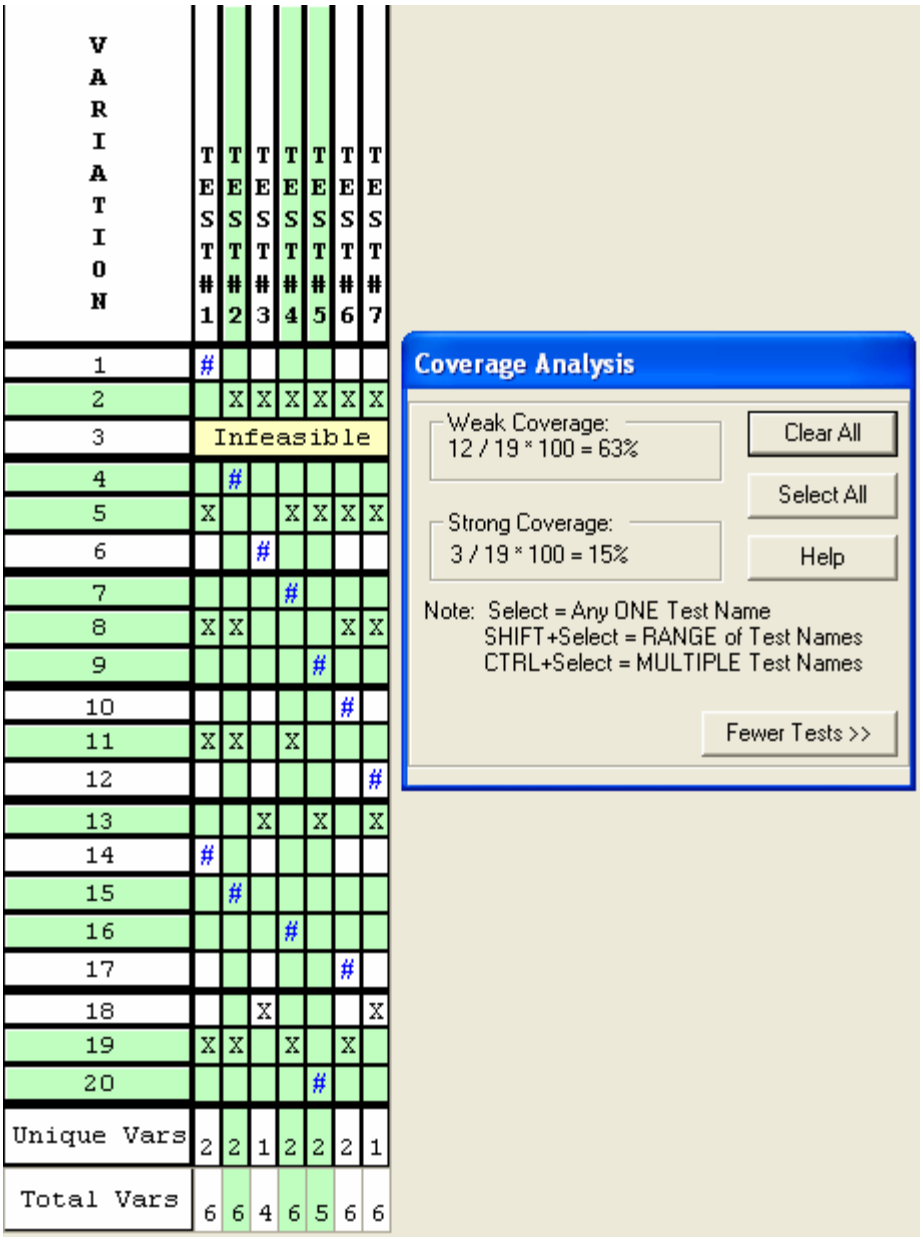
4. Since BenderRBT produces the appropriate set of test cases to cover 100% of the functionality for the system under test, once each of these test cases is successfully executed, then test execution is complete. BenderRBT also provides the ability to show how much functional coverage the execution of 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.



VARIATION	TEST#1	TEST#2	TEST#3	TEST#4	TEST#5	TEST#6	TEST#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%

Clear All

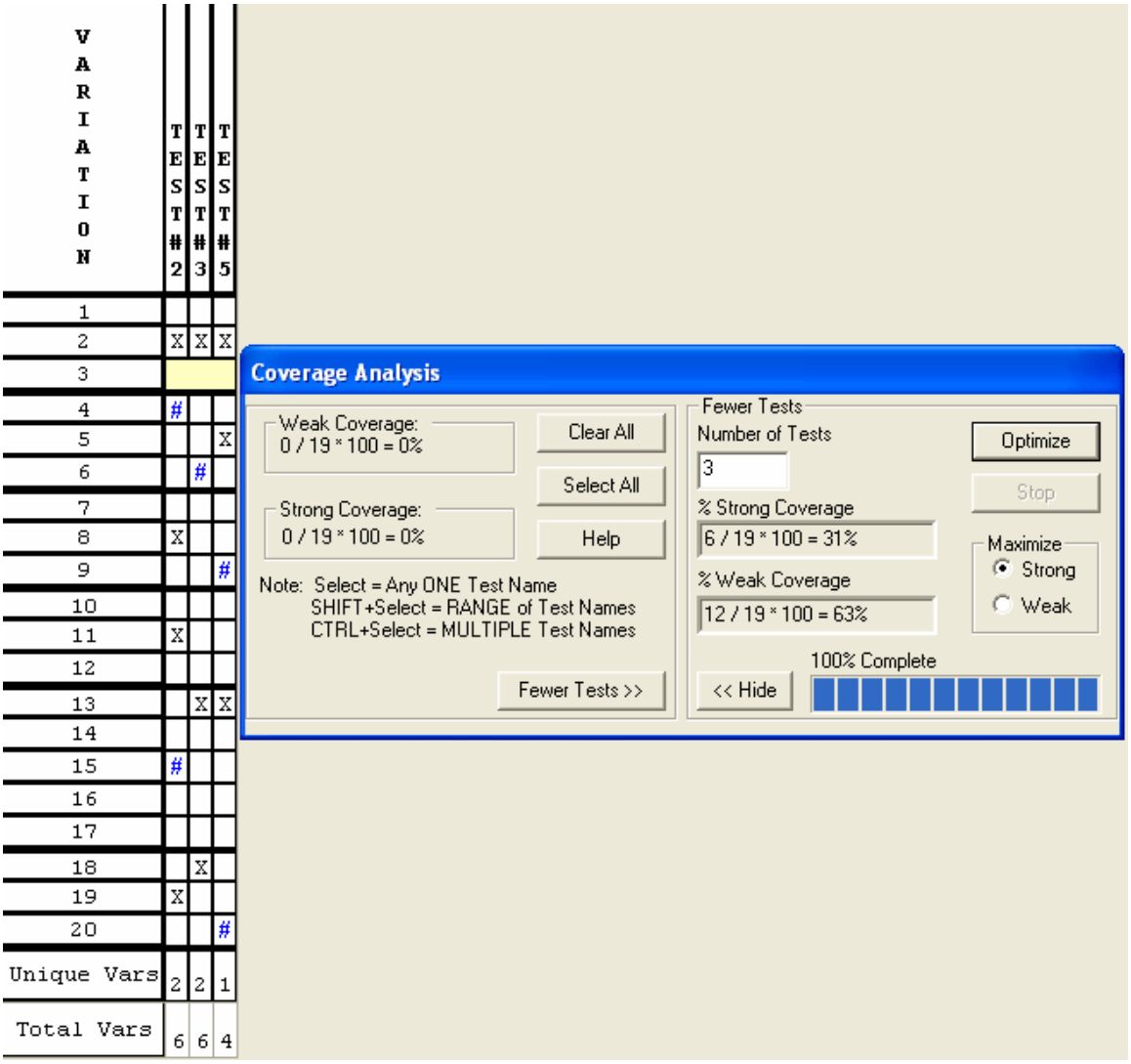
Select All

Help

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

Fewer Tests >>

Coverage Analysis Dialog



VARIATION	TEST #2	TEST #3	TEST #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

Fewer Tests Dialog

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