

Using IBM/Rational with SOAtest

SOAtest test cases can be mapped to specific requirements or test plans in Rational TestManager. Once a test case is mapped to a test plan, the testing of the requirement becomes as simple as pressing a button inside TestManager. SOAtest test cases can be run from inside Rational TestManager and the results are reported back to TestManager.

Additionally, Parasoft SOA Quality Solution provides native support for the IBM WebSphere MQ API; tests JMS, EJB and web services on WebSphere; and provides WebSphere JMX performance monitoring which exposes the metrics typically available on WebSphere Tivoli.

For WebSphere ESB, you can also visualize and trace the intraprocess events that take place as part of the transactions that are triggered by the tests, and then dissect them for validation. In addition to providing visibility into the system's intermediate messages, this allows you replay transactions directly from the solution and verify that the monitored functionality continues to work as expected. As a result, test engineers gain the ability to identify problem causes and validate multi-endpoint, integrated transaction systems that have been traditionally handled only by specialized development teams. For details, see [Monitoring IBM WebSphere ESB](#).

Using SOAtest with Rational TestManager

The following sections describe how to integrate SOAtest into Rational TestManager. First, SOAtest must be configured to have access to various TestManager jar files. Second, a reusable test script is created. Third, test plans are created using the reusable test scripts.

Sections include:

- [Configuring the SOAtest Classpath](#)
- [Configuring a Reusable SOAtest Test Script in TestManager](#)
 - [Creating a new script using the command line execution adapter](#)
 - [Creating a test plan using an existing test script](#)

Configuring the SOAtest Classpath

Before you start, make sure to include all required JAR files to the classpath. Required JAR files include `rtjavatestserver.jar`, `rttssjava.jar`, `rttseajava.jar`, and `rational_ct.jar`.

You can find the jar files in the following default locations:

```
E:\Program Files\Rational\Rational Test\rtjavatestserver.jar
E:\Program Files\Rational\Rational Test\rttssjava.jar
E:\Program Files\Rational\Rational Test\tsea\rttseajava.jar
E:\Program Files\Rational\Rational Test\QualityArchitect\rational_ct.jar
```

There are two ways you can add JAR files:

- Add JAR files in SOAtest's classpath (recommended): For more information on adding JAR files to SOAtest's classpath, see [System Properties Settings](#).
- Add JAR Files to the system classpath:
 - For Windows:
 1. Right-click **My Computer** and select **Properties** from the shortcut menu.
 2. Select the **Advanced** tab and click the **Environment Variables** button.
 - If a CLASSPATH variable is not yet created, click the **New** button either for **User variables** or **System variables** and enter CLASSPATH for the **Variable name** and `< path to the jar files >` for **Variable value**.
 - If a CLASSPATH variable is already created, double-click the CLASSPATH variable and append `< path to the jar files >`
 3. Click **OK**.
 - For UNIX: Please refer to the Oracle Java documentation for setting the JAR files to the classpath.

Configuring a Reusable SOAtest Test Script in TestManager

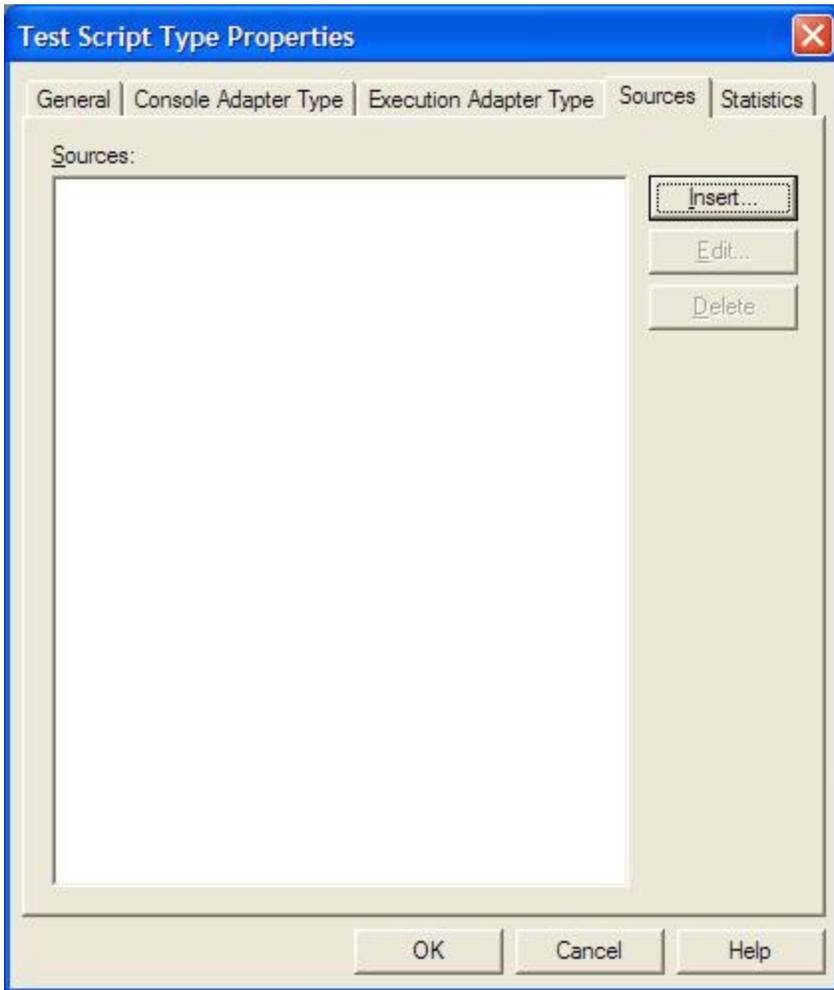
Creating a new script using the command line execution adapter

1. In TestManager, select **Tools> Manage> Test Script Types**.
2. Click **New** to select a new Test Script Type, then enter a **Name**.
3. In the **ExecuteAdapterType** tab, choose the **Use command line execution adapter** radio button and manually enter the following in the Execution command line field:

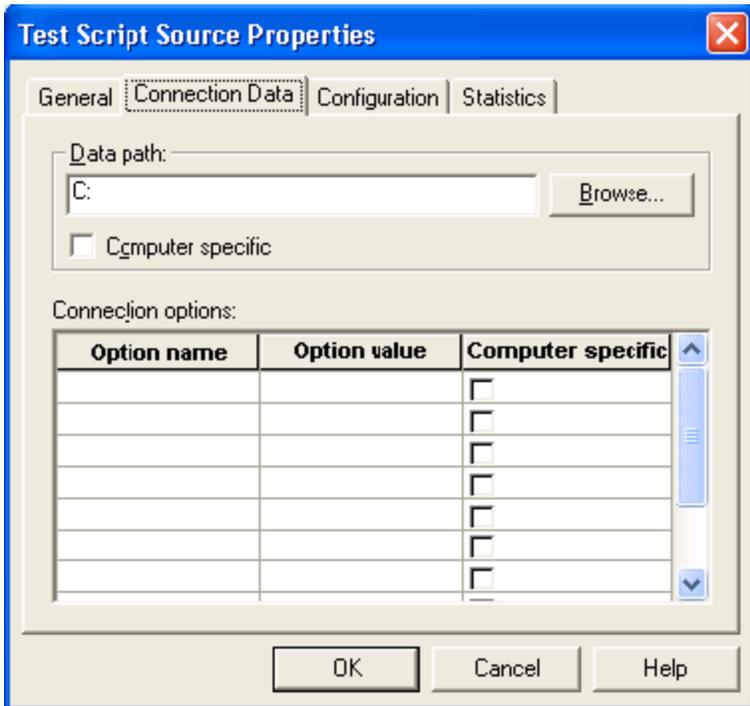
```
<path to SOAtest's execution> -config configName -resource relPathToTest [-testname {testname}] -testManagerVerbose
```

- For example:

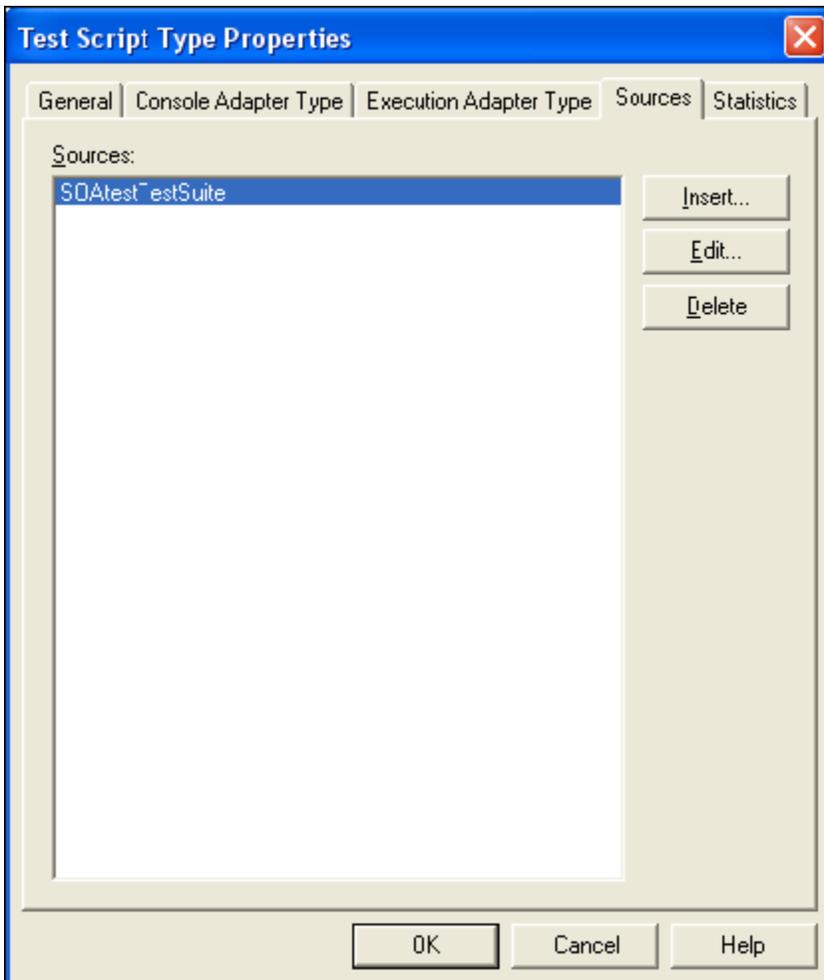
```
"C:\Program Files\Parasoft\SOAtest\[SOAtest version number]\soatestcli.exe" -config "user://Example Configuration" -resource myProject/myTest.tst -testManagerVerbose
```
 - For additional command line options, see [Testing from the Command Line Interface - soatestcli](#).
4. Select the **Sources** tab in the Test Script Type Properties dialog and click the **Insert** button.



5. After clicking **Insert**, a **New Test Script Source** dialog opens. Enter a **Name** in the **General** tab for the new Test Script Source and then select the **Connection Data** tab.
6. In the **Connection Data** tab, select the data path to a folder that contains the SOAtest test suites (.tst) then click **OK**.

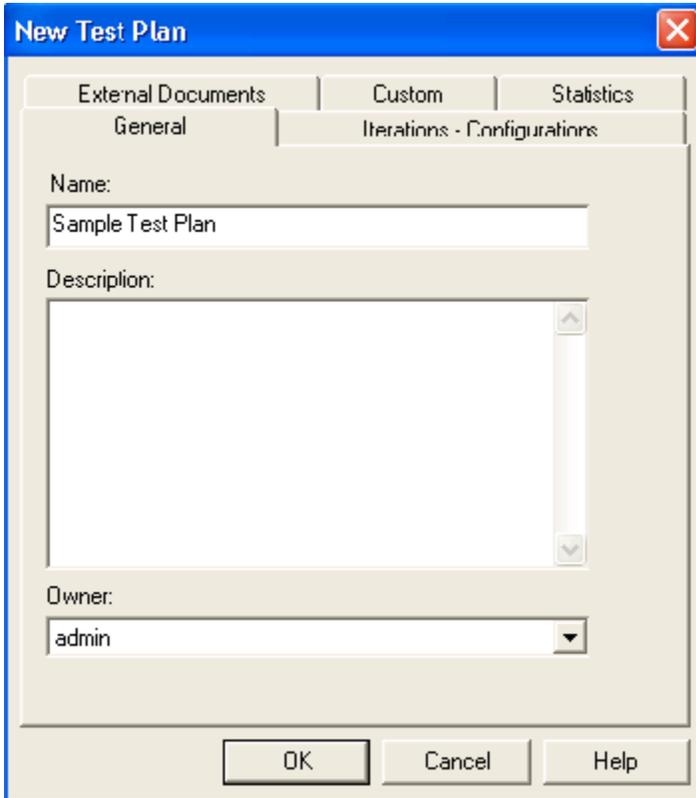


7. After clicking **OK** in the Connection Data tab, the new source appears in the Test Script Type Properties dialog. Click **OK** to finish.

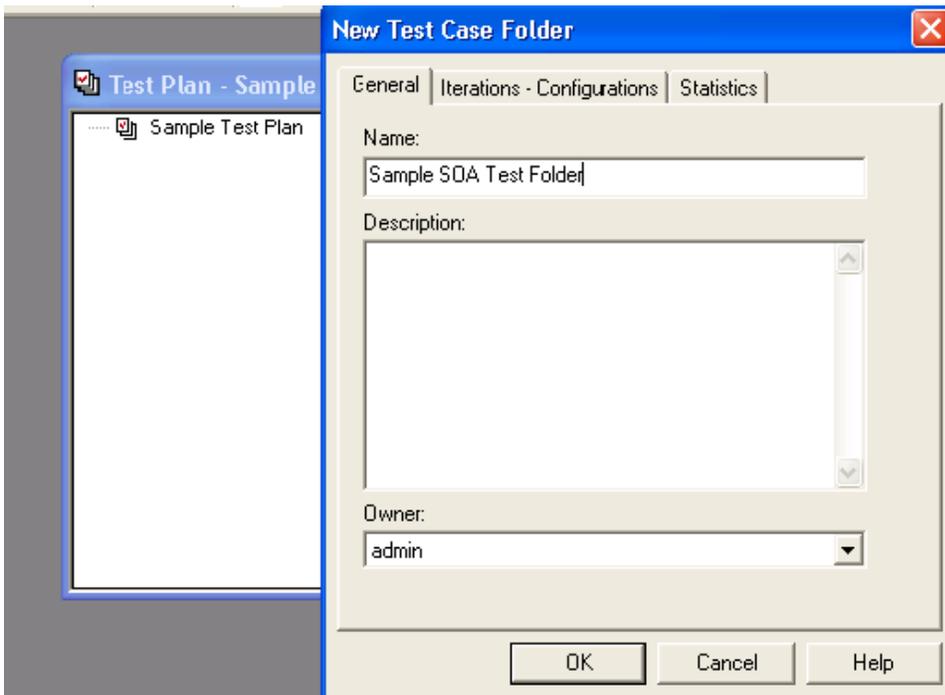


Creating a test plan using an existing test script

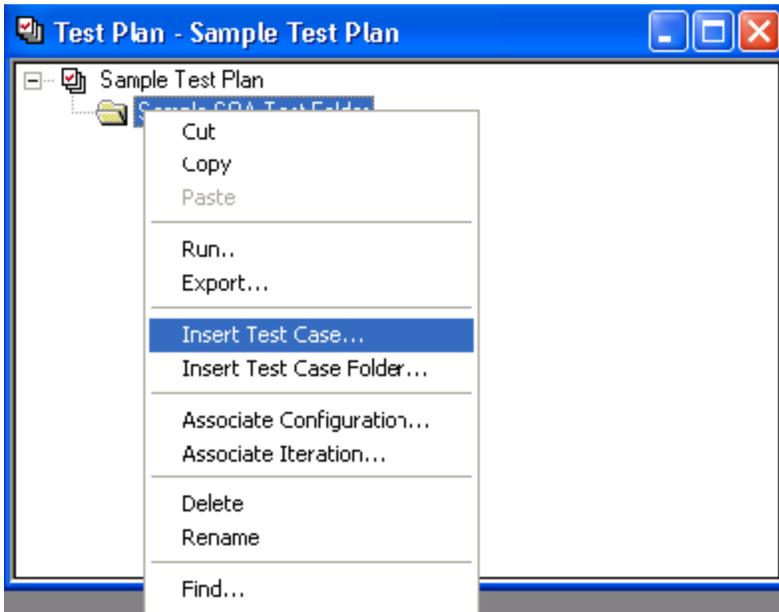
1. In the left GUI panel, select the **Planning** tab, right-click the **Test Plans** node and select **New Test Plan** from the shortcut menu. The New Test Plan dialog displays.



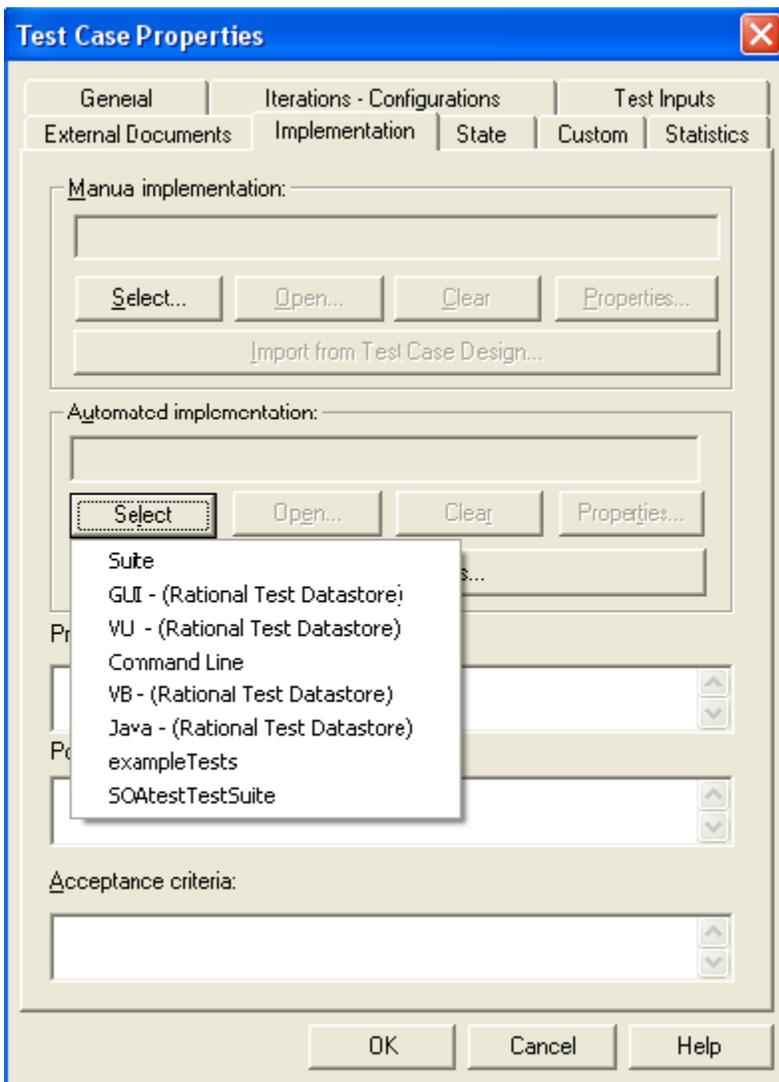
2. In the **New Test Plan** dialog, enter a **Name** for the test plan in the **General Tab** and click **OK**. The newly created test plan displays in the left GUI panel under the Test Plans node.
3. Double-click the newly created test plan to open it. Right-click the test plan and select **Insert Test Case Folder** from the shortcut menu. A **New Test Case Folder** dialog displays.



4. In the **New Test Case Folder** dialog, enter a **Name** and click **OK**. The new test case folder displays in the test plan.
5. Right-click the new test case folder and select **Insert Test Case** from the shortcut menu.

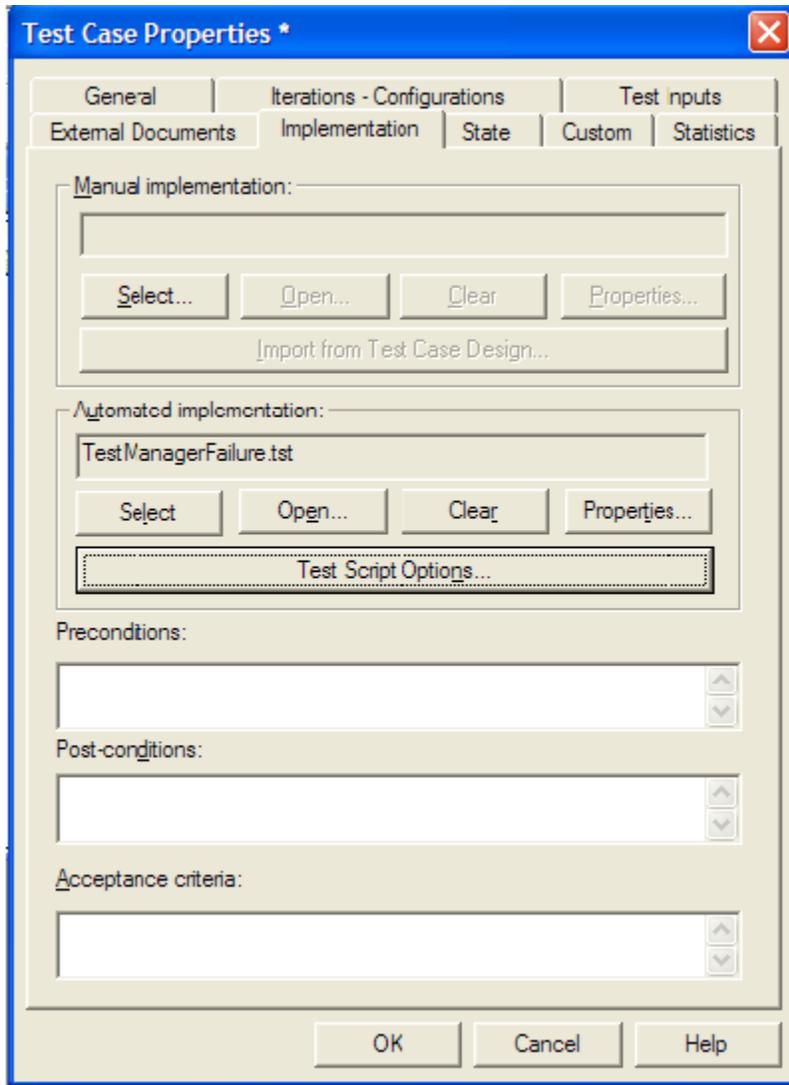


- In the **Implementation** tab of the **New Test Case** dialog, click the **Select** button for **automated implementation**. Choose the desired test script from the drop-down menu (in this case, the test script created previously in [Creating a new script using the command line execution adapter](#)).

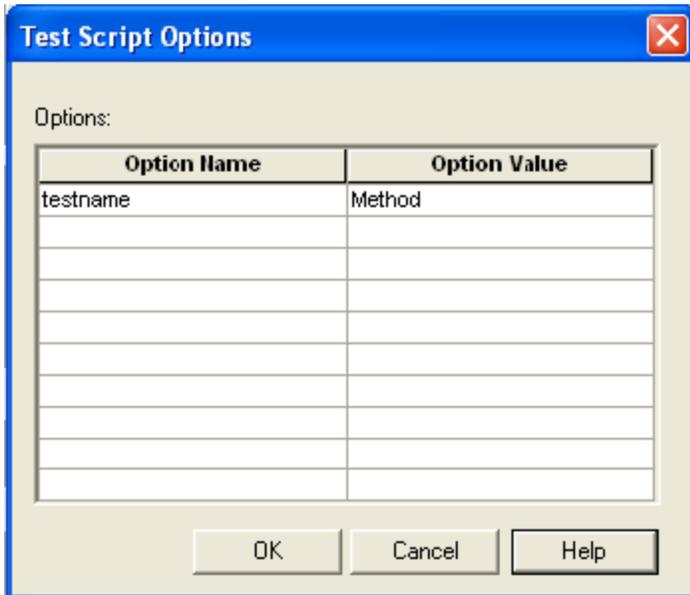


- TestManager now prompts you to select a script file; browse and select any SOAtest project file (.tst) you would like to test.

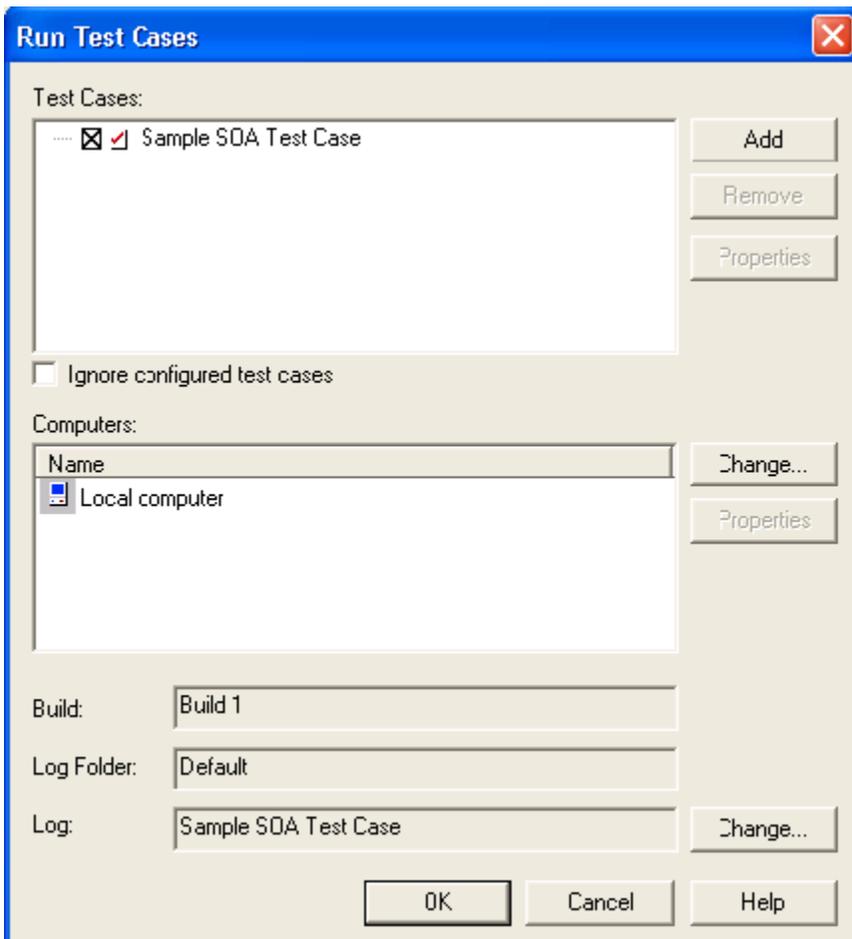
8. Click the **Test Script Options** button.



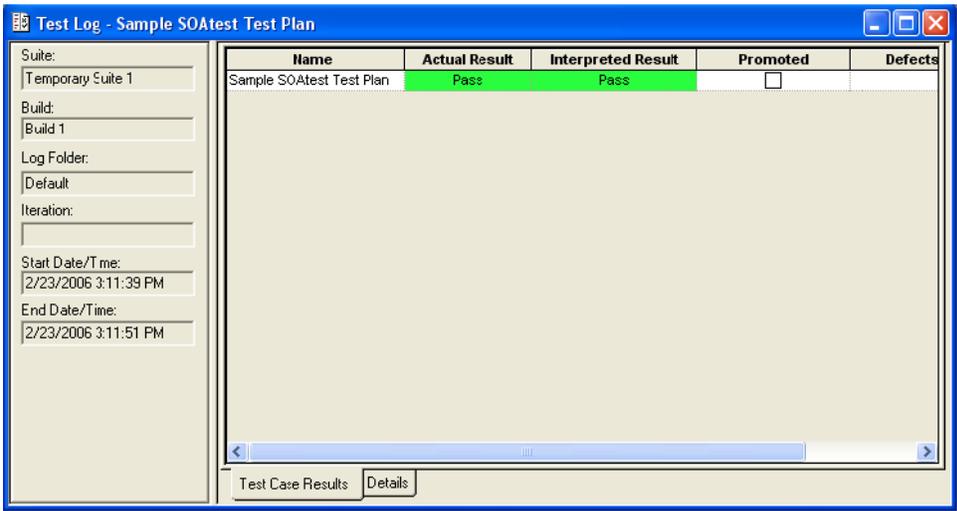
9. Type **testname** as the **Option Name**. Type the name of the test you want to run as the **Option Value**. In the following example, the variable 'testname' was mapped to the value 'Method'. If you included the `-testname {testname}` option in the command line execution adapter, SOAtest will only run the test named 'Method'.



10. In the Test Plan dialog, right-click the new test case and choose **Run** from the shortcut menu. A Run Test Cases dialog displays.



11. Click **OK**. TestManager will now run the test script with the project file selected in step 7. If the Test Suite succeeds, you should see the results in the Test Log.



12. Click the **Details** tab in the Test Log window and expand the nodes to view the properties of the User Defined node. SOAtest reports the results as a user-defined event.

