

Using Microsoft with SOAtest

Microsoft .NET WCF (Windows Communication Foundation) allows for the creation of rich web services. However, due to proprietary service bindings (protocols) and complexities within the WS-* standards that it supports, .NET WCF poses testing challenges. Given this, the Microsoft environment truly necessitates a "SOA-Aware" and "WCF-Aware" testing solution such as Parasoft SOA Quality Solution. For details, see [.NET WCF HTTP](#), [.NET WCF TCP](#) and [.NET WCF Flowed Transactions](#).

In addition, Parasoft SOA Quality Solution integrates with Microsoft Visual Studio Team System (VSTS) Edition for Testers. Integration with MS VSTS allows for the management and execution of test projects, and obtaining results directly within Visual Studio, thereby streamlining testing for users of VSTS within their Visual Studio native environment.

Microsoft Visual Studio Team Edition for Software Testers is part of the Visual Studio Team System (VSTS) platform. It allows for creating, managing, sharing and executing tests within Visual Studio. Parasoft SOAtest includes the capability to integrate with MS VSTS so that SOAtest functional tests can be managed and executed within the Visual Studio IDE and the results of execution are directly displayed within Visual Studio. This enables teams that use the VSTS platform to integrate SOAtest into their process in order to streamline their joint Microsoft and Parasoft solutions, and to further the return on investments made in the existing infrastructure.

Using SOAtest with Microsoft Visual Studio

The following sections describe how to integrate Parasoft SOAtest with MS VSTS for Software Testers. These instructions were written for VSTS 2005.

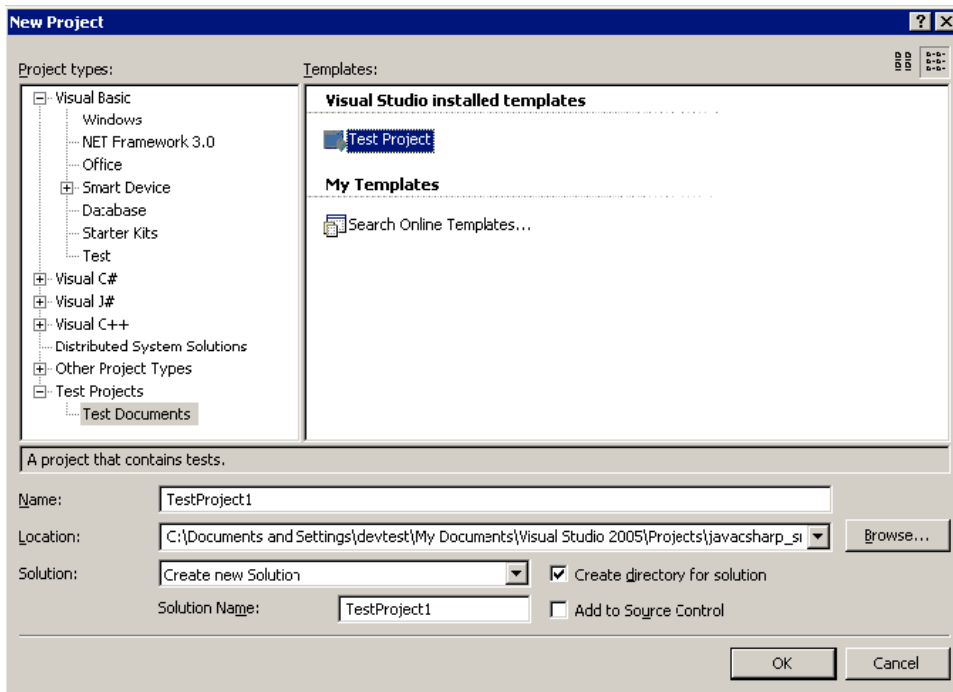
Sections include:

- [Creating a Test](#)
- [Executing Tests](#)
- [Viewing the Results](#)

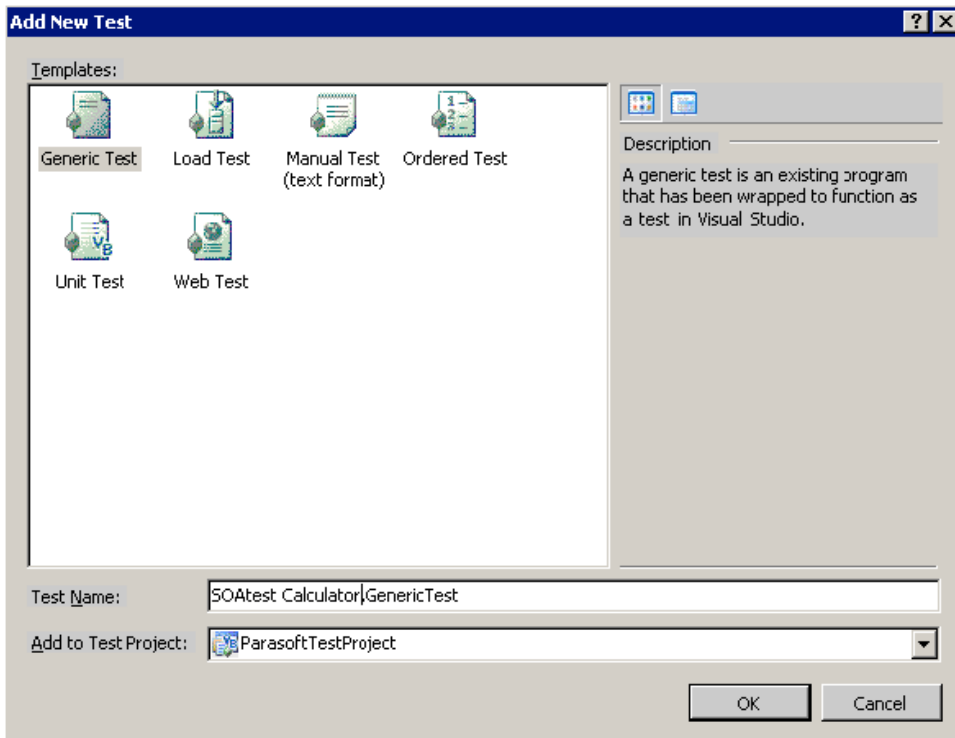
Creating a Test

This section describes the process of creating a new VSTS test case within Visual Studio and associating it with an existing Parasoft SOAtest project (.tst) file. To create a new VSTS test case:

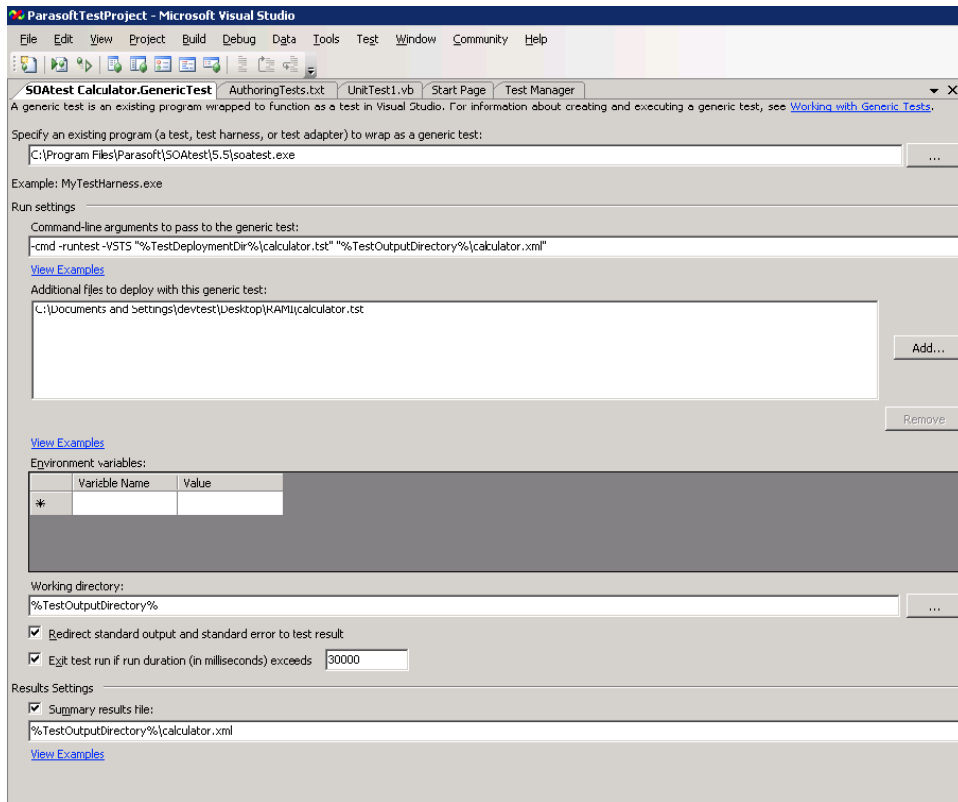
1. Create a new Visual Studio test project or open an existing one. To create a new test project in Visual Studio:
 - a. Navigate the menu **File> New> Test Project...**
 - b. Select **Test Projects** under the **Project types** tree on the left.



2. Create a new test by selecting **Test> New Test...**



3. Select **Generic Test** from the **Templates** field and then enter a name in the **Test Name** field.
4. Configure the test with an existing Parasoft SOAtest project file (.tst file):
 - a. In the existing program field at the top, specify the path to the soatestcli executable. For example: C:\Program Files\Parasoft\SOAtest\[soatest version number]\soatestcli.exe
 - b. Under **Run settings** for the command line arguments field, specify the following arguments -config "user://Example Configuration" -visualStudio xmlReport -resource testSuite -report htmlReport
For example:
-config "user://Example Configuration" -visualStudio C:\report\calculator.xml -resource calculator_project\calculator.tst -report C:\report\calculator.xml.details.html
The example above works for calculator.tst in a project called calculator_project in the workspace. The HTML file name should be xml file name + .details.html.
 - c. In the **Additional files to deploy with this generic test:** field, browse to and add your SOAtest project file.
 - d. Under the **Results Settings** section, select the **Summary resultsfile** option and specify the same xml results file name as in the argument added in step **b** above.



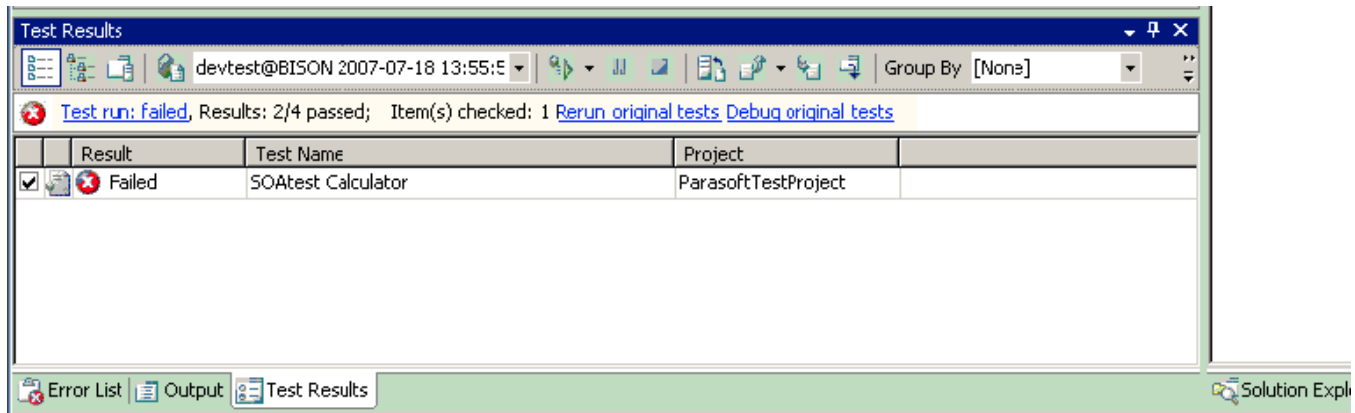
- e. Save the test configuration (press **CTRL+S**, or select **File> Save**).

Executing Tests

Now that the Visual Studio generic test has been associated with a SOAtest project file, you may now run it. You can do so by running the entire test project from the tool bar or right-clicking on that particular test in the **Test View** and selecting **Run Selection**.

Viewing the Results

Once execution of one or more SOAtest tests finishes, a summary of results is shown in the **Test Results** window within Visual Studio:



Notice the results count (2/4 in the figure above). This means that 2 SOAtest test cases out of 4 have passed. A test case could be a SOAP Client test, or another tool test within a SOAtest test suite.

To view details, double-click on the test rule row (on the test name in the figure above)

The **Inner Test Results** area displays pass/fail results for each SOAtest test case along with the error message. The **Summary File** area shows an HTML rendered view of the execution report, similar to the report that SOAtest can generate in a standalone execution.

Common Results

Test Name: SOAtest Calculator
 Result: Failed
 Duration: 00:00:21.4848394
 Computer Name: BISON
 Start Time: 7/18/2007 1:55:59 PM
 End Time: 7/18/2007 1:56:20 PM

Generic Test Common Results

Internal Test Name: Test Suite: Calculator Tests
 Exit Code: 0
 Aborted: No
 Detailed Results File:

Inner Test Results

Test Name	Test Results	Error Message	Test Result Details
Test 1: add(add)	Passed		C:\Documents and Settings\devtest\My Documents\Visual Studio 2005\Projects\ParasoftTestProject\TestResults\devtest_BISON 2007-07-18
Test 1: multiply(multiply)	Failed	Changed element text content value	C:\Documents and Settings\devtest\My Documents\Visual Studio 2005\Projects\ParasoftTestProject\TestResults\devtest_BISON 2007-07-18
Test 2: divide(divide)	Passed		C:\Documents and Settings\devtest\My Documents\Visual Studio

Summary File

PARASOFT
We make software work.

Test Execution Details

Projects Summary

Project	Tests Run	Succeeded	Failed	%
X calculator.tst	3	2	1	66.0%
X Total	3	2	1	66.0%

Test Results

devtest@BISON 2007-07-18 13:55:59 | Group By [None] | [All Columns] | <Type keyword>

Test run: failed, Results: 2/4 passed; Item(s) checked: 1 [Rerun original tests](#) [Debug original tests](#)

Result	Test Name	Project
Failed	SOAtest Calculator	ParasoftTestProject

i Reporting Results from the Command Line

You can also report results to VSTS using the `-visualStudio` cli option. For example: `soatestcli.exe -config "team://MyConfiguration" -visualStudio -resource CLI_project\secondtest.tst`