Migration Guide for Existing SOAtest and WebKing Users

This topic is a general migration guide for users of WebKing or of SOAtest version 5.x and earlier. Sections include:

About This Migration Guide

This guide is designed to help users of WebKing or of SOAtest version 5.x and earlier get up and running in the latest version of SOAtest as rapidly as possible.

This Migration Guide is meant to be used by users already familiar with SOAtest or WebKing. New users should review the SOAtest Tutorial first.

Setting Up Projects for Existing Tests

Since the SOAtest interface has been integrated into the Eclipse framework since version 6.0, it now follows the Eclipse framework hierarchy for managing your test assets. You no longer need to open .tst files one at a time. Instead, you can manage all your .tst files in projects within a workspace.

- A workspace corresponds to a directory on the local machine. SOAtest will ask you for the desired location of workspace at startup and will remember that location in future runs. When you start the SOAtest 9.x and later, an Eclipse workspace is automatically created in `<user_home_dir>/parasoft/workspace`. For instance: `/home/username/parasoft/workspace` (Linux), `C:\Users\username\parasoft\workspace` (Windows).
- A workspace can contain multiple projects, each of which correlates to a directory inside the workspace on the local machine. The project can contain multiple .tst files along with any related files and artifacts such as data source Excel spread sheets, keystores, etc.
- The .tst file inside a project serves the same function as what was called a "project file" in previous releases.

Choosing an Appropriate Project Setup Strategy

In the following sections, we will go through several ways of creating new projects that may be useful to existing SOAtest or WebKing users. Other ways of creating new projects, (e.g. from a WSDL), can be found in the SOAtest Tutorial.

To ensure that tests can easily be shared across your organization, a designated team member—usually a team lead or manager—must decide which project setup strategy to use. The entire team should then adopt the same strategy.

In these situations	Use this strategy
Your tests are stored in a source control system	Creating Projects from Tests Under Source Control
Your tests are NOT stored in a source control system and you want to copy your old tests into a new location on your file system	Copying Tests to a New Location
Your tests are NOT stored in a source control system and you want the existing files to remain in the same location on your file system	Leaving Tests in the Original Location

Auto-save to Newer Formats in SOAtest 6.0 and Later

Once a file is opened within SOAtest 6.0 or later, it is automatically saved in a new format that cannot be opened in earlier versions of SOAtest or WebKing.

Creating Projects from Tests Under Source Control

By default, SOAtest 9.x ships with CVS source control support. Support for additional source controls can be added by providing appropriate plugins to Eclipse.

To create a project consisting of test suites that are checked into source control:

1. Choose **File> Import**.
2. In the window that opens, expand the folder that corresponds to your source control system (e.g., **SVN** or **CVS**).
3. Select **Project(s) from <name of source control>** then click **Next**.

4. Enter the necessary Repository Location Information for the source control folder containing your tests, then click **Finish**.
5. After the project is available in your workspace, add the `.project` and `.parasoft` files into source control. They will be visible in the Navigator view and should be shared by the entire team.
    - Do NOT add the `.metadata` folder into source control.

Creating Projects from Tests That Are Not Under Source Control

We strongly recommend copying the old tests into your new workspace. Doing so will preserve your old tests in a manner analogous to backing up your hard drive. This procedure is explained in Copying Tests to a New Location.

As a second option, you can use the Project from Existing SOAtest or WebKing Test Suites wizard for tests not stored in a source control system. This will cause the original files to appear within your workspace—but it will allow them to remain in the same location on your file system. This procedure is explained in Leaving Tests in the Original Location.

Copying Tests to a New Location

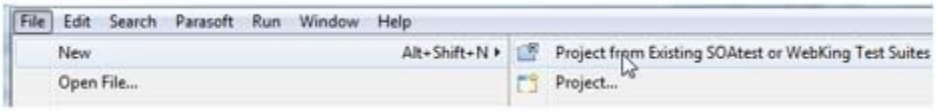
To create a project that copies existing test suites to a new location on the file system:

1. Choose **File> New> Project**.
2. In the window that opens, expand **General**, select **Project**, then click **Next**.
3. Specify a name for your project (which will contain multiple tst files), then click **Finish**. This will create an empty folder in your workspace.
4. Choose **File> Import**.
5. In the window that opens, expand **General**, select **File System**, then click **Next**.
6. In the **From directory** field, navigate to the directory containing your tests.
7. In the **Into folder** field, select your project folder from Step 3, then click **Finish**.
8. (Optional, Strongly Recommended) Obtain a source control system, and add the entire project, the `.project` folder, and the `.parasoft` files into source control. They will be visible in the Navigator view and should be shared by the entire team.
    - Do NOT add the `.metadata` folder into source control.

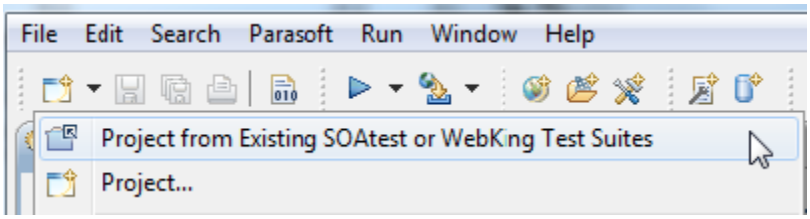
Leaving Tests in the Original Location

To create a project that leaves existing test suites at the same location on the file system:

1. Open the **New Project Wizard** by completing one of the following options:
    - Select **File> New> Project from Existing SOAtest or WebKing Test Suites**.



    - Open the pull-down menu for the **New** toolbar button (top left) then choose **Project from Existing SOAtest or WebKing Test Suites**.



2. In the wizard that opens, enter a project name then enter or browse to the root directory for the existing test suites.
3. Click the **Finish** button. The tests you selected will display in the Test Case Explorer.

Using a Team Project Set File to Share a New Project Across the Team

Once a team member creates a project, that team member can create a Team Project Set File (.psf) that can then be shared with other members of the team. This allows every team member to create their Eclipse Projects in the same uniform way. This is a necessary step for importing tasks from the automated nightly test process.

To create a Team Project Set File for a project created from CVS, complete the following:

1. Select **File> Export**. The **Export Wizard** displays.
2. Within the **Export Wizard**, select **Team> Team Project Set**, and then click the **Next** button.
3. Select the projects to be included in the Team Project Set File by selecting the corresponding check boxes.
4. Enter the location where the Team Project Set File will be saved and click the **Finish** button.

To create a Project from the Team Project Set File, complete the following:

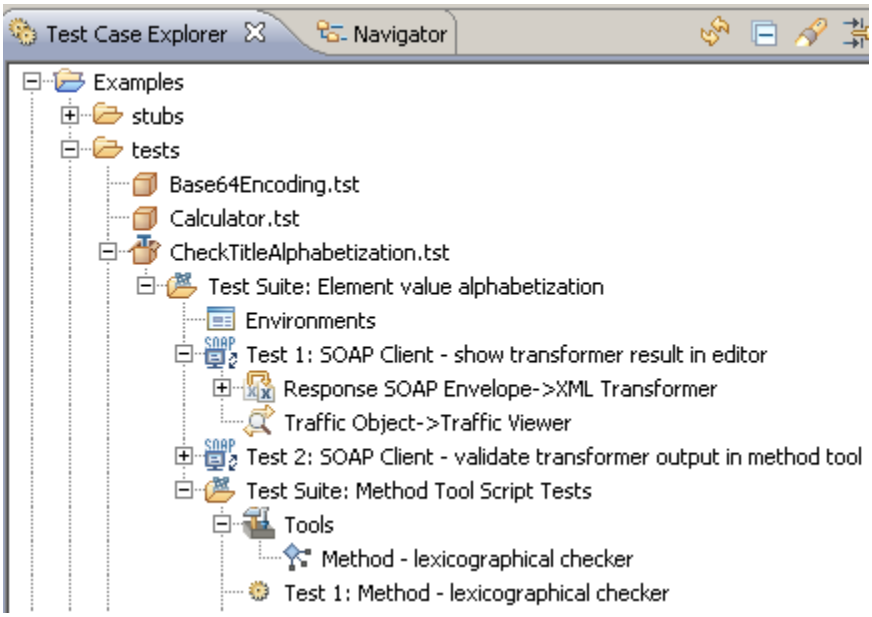
1. Select **File> Import**. The **Import Wizard** displays.
2. Within the **Import Wizard**, select **Team> Team Project Set**, and then click the **Next** button.
3. **Browse** to the desired Team Project Set and click the **Finish** button. The tests that you selected display in the Test Case Explorer.

Familiarizing Yourself with the SOAtest 9.x and Later Interface

Parasoft SOAtest is based on the Eclipse IDE and has a different look and feel than older versions. Except for the changes outlined below, however, the user interface design layout, forms and settings have largely remained unchanged and should remain familiar to existing users.

Test Case Explorer

The Test Case Explorer can have multiple Eclipse projects open at the same time. Each project can have multiple Test Suites open at the same time. In previous versions of SOAtest, only one Test Suite could be open at any given time.



Test Case Explorer Menu Buttons

At the top right corner of the Test Case Explorer are the following menu buttons:

- **Refresh:** Click to refresh the contents of the Test Case Explorer.
- **Collapse All:** Click to collapse all of the nodes within the Test Case Explorer.
- **Search:** Click to perform a search for any node (i.e. Test Suites, tests, chained tools, etc.) within the Test Case Explorer. After clicking the Search button, the following options display:
    - **Containing:** Enter the text or string contained within the test.
    - **Within the whole tree:** Select to search for the specified text within the whole tree.
    - **Within the selected node:** Select to search for the specified text within the selected node.
    - **Wrap around:** Select to perform a search on wrap around text.
    - **Case sensitive:** Select to perform a case sensitive search.
- **Filter:** Select to hide specified projects or tests within the Test Case Explorer.
- **Statistics:** Select to display statistics (i.e. number of tests Passed, Failed, Errors, Skipped, Run) next to each test suite node within the Test Case Explorer.

Editors

Opening Editors with a Double or Single Click

In previous versions, if you wanted to open the configuration panel for a test node (e.g., an "Editor"), you would select that node in the Tests tab. With SOAtest 9.x, you double-click on an item's Test Case Explorer node to display its Editor.

If you want to change the default double-click behavior to single-click, complete the following:

1. Select **Window> Preferences**. The **Preferences** dialog displays.
2. Within the Preferences dialog, select **General** on the left, and change the **Open mode** from **Double click** to **Single click** within the right GUI panel.
3. Select **General> Editors**, enable **Close editors automatically** and then click the **OK** button.

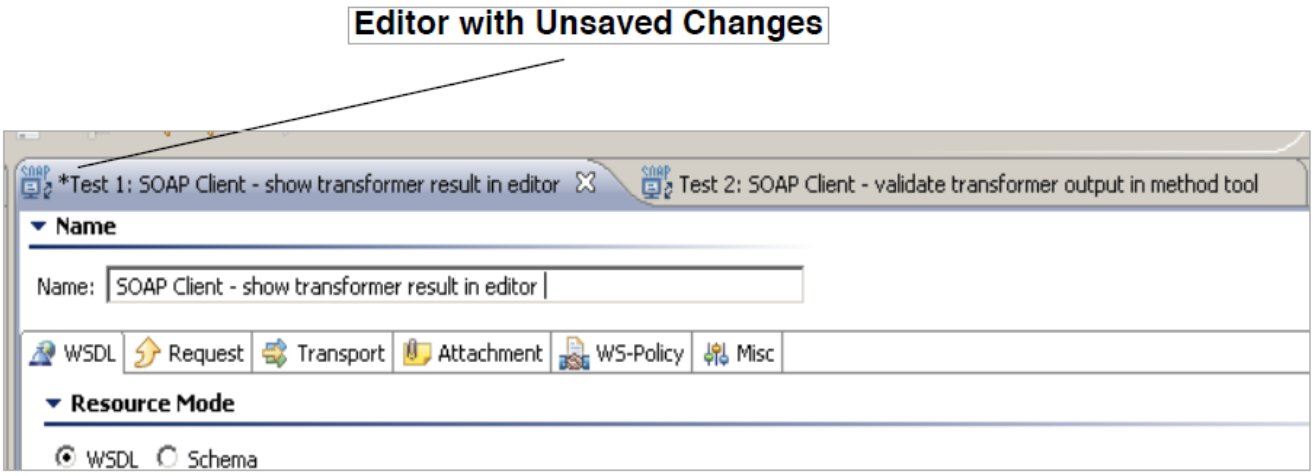
You will now be able to open editors based on a single click.

Opening Multiple Editors

In previous versions of SOAtest and WebKing, only one Editor could be open at once. In SOAtest 9.x, multiple Editors can be open simultaneously.

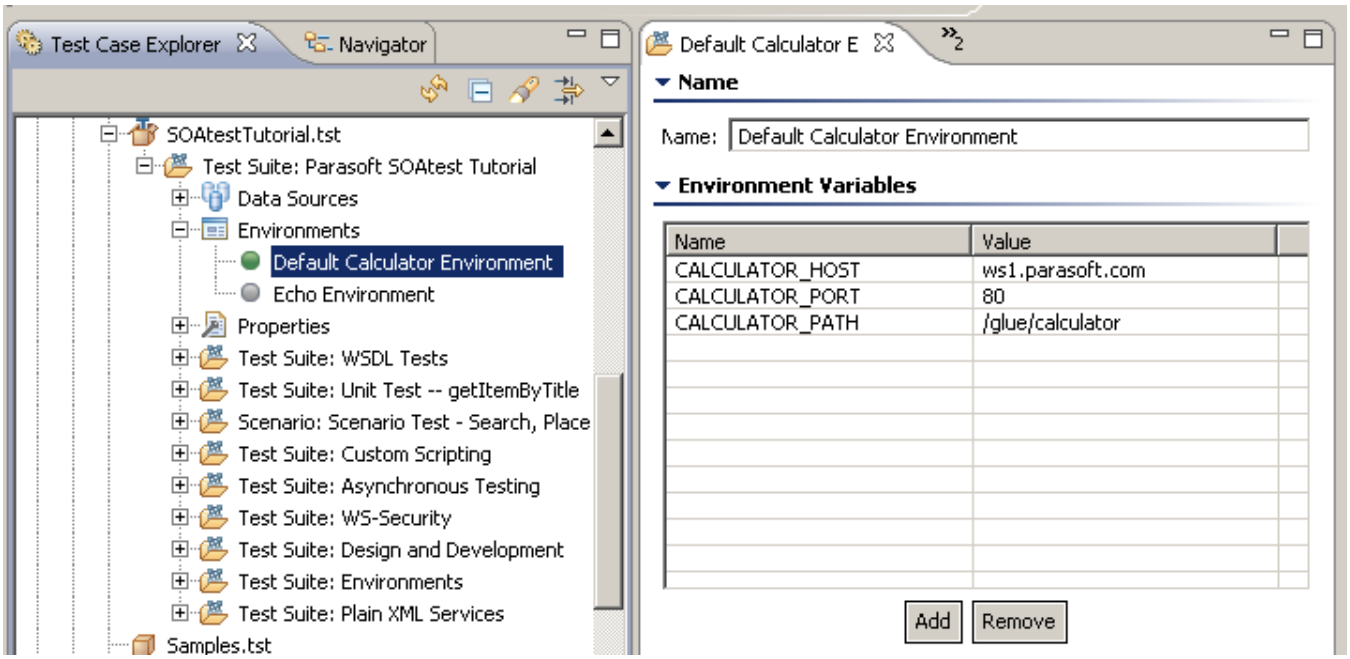
Saving Changes in Editors

When an Editor is modified in SOAtest 9.x and later, an asterisk "*" displays on the Editor tab denoting that the Editor is now "dirty." Modifications to the Editor must be explicitly saved using the **Save** toolbar button or the **Ctrl-S** keyboard shortcut.



Environments

In previous versions of SOAtest (5.x and earlier), environments were displayed in a separate tab below the Tests tab. Environments are now part of the tree view in the Test Case Explorer.



Running a Test

To run a test, you can right-click on the test's node and select **Test Using 'Example Configuration'** from the shortcut menu.' Alternatively, you can press **F9** on your keyboard, or click the **Test** toolbar button.

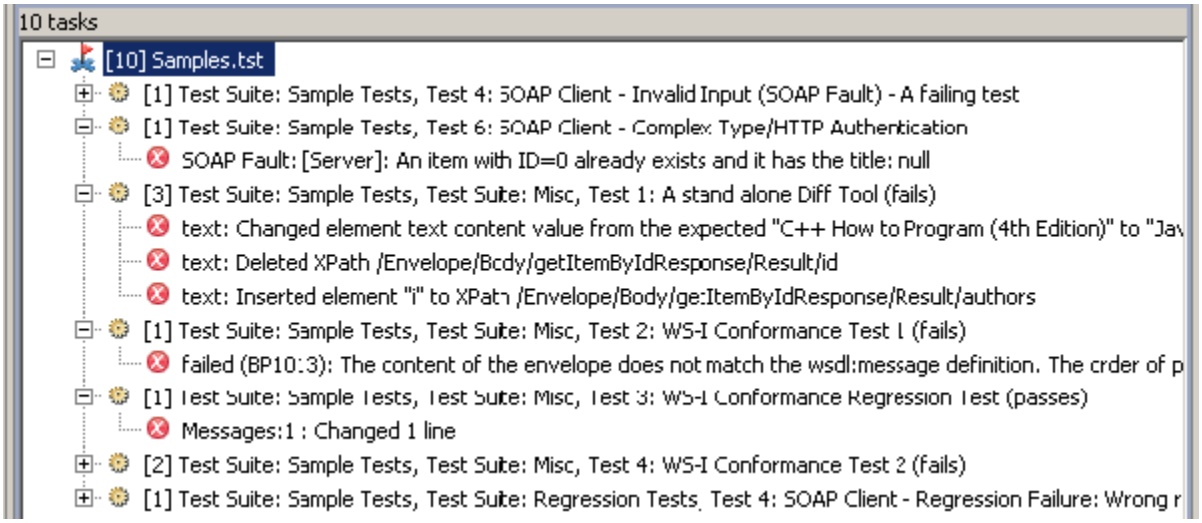
Saving Test Suite Files

In previous versions of SOAtest and WebKing, you had to explicitly save test suite (`.tst`) files. In SOAtest 6.x and later, user actions in the Test Case Explorer are automatically saved. For instance, adding new Test to the Test Case Explorer will be automatically saved.

Once tests are saved in the latest version of SOAtest, they cannot be opened in older versions of SOAtest.

Quality Tasks view and Console View

Failures that occur during the test execution now display in the Quality Tasks view. What was previously displayed in the Messages Log now displays in the Console view.



Source Control Integration

If you have the appropriate source control plugins installed into the Eclipse environment, your Test Suites can now be checked into source control directly as follows:

- Right-click the Test Suite node and select **Team> Commit** from the shortcut menu.

To check in new projects, complete the following:

- Right-click on the Project node and select **Team> Share Project** from the shortcut menu.

Setting up the Automated Nightly Process using CLI

To set up an automated nightly build from the Command Line, complete the following:

1. Start SOAtest on your test machine, then create a workspace containing all the projects and test suites that you wish to run as part of your nightly testing process. For more information, see the Setting Up Projects for Existing Tests section above.
2. Configure SOAtest's preferences with any global settings that are required for your tests. To open SOAtest preferences, choose **Parasoft> Preferences**. If the test suites in your workspace were imported from source control, then you should configure the **Parasoft> Source Controls** settings.

3. (Optional) Create a test configuration to use for your nightly test run. Test configurations have settings that affect the way in which your tests are executed. SOAtest ships with a test configuration named **Example Configuration** that you can use if you do not wish to create your own. Your test configurations can be managed by choosing **Parasoft> Test Configurations**. If the projects in your workspace were created from source control, then you should click the **Common** tab in your test configuration then enable the **Update projects from source control** option.
4. (Optional) Create Local Settings Files - Options. This are text files that can be used to control settings for reports, email, Report Center, Team Server, license server, authorship, and source control.
5. Schedule a daily process to invoke SOAtest using the desired command line options. This can be done using a job scheduler mechanism such as Windows Task Scheduler or Unix cron. For example, to run all projects in your workspace you can use the following command:

```
soatestcli.exe -data "c:\mySOAtestWorkspace" -showdetails -config "user://Example Configuration" -report "c:
\mySOAtestReports" -publish -localsettings c:\mySOAtestWorkspace\mylocalsettings.properties"
```

The `-publish` argument will add reports to DTP so that test and analysis data can be merged, correlated, and analyzed to expose deeply hidden defect patterns. As the DTP processes data, it creates actionable findings that can be downloaded and imported into your IDE (requires DTP 5.3.x or later).

You can also use the `-publishteamserver` option to publish reports to Team Server, which provides backwards compatibility with Concerto and older versions of DTP.

For a detailed list of changes, see the topic on Command Line Interface Migration.

HP Quality Center Integration

There have been upgrades to the HP QC Integration from 6.2 to 9.x. The steps to connect the two products must be redone in order to ensure that the correct behavior is continuing.

Deprecated Features

- **Load Testing:** This is now available in a separate installable called Parasoft Load Test. The current version will allow you to run your existing SOA and web load tests as well as create new ones. It also allows you to load test complete end-to-end test scenarios—from the web interface, through services, to the database. Every protocol and test type available in Parasoft SOAtest is supported in Parasoft Load Test.
    - *Parasoft Load Test includes the full SOAtest product, so if you are interested in both functional testing and load testing, you should install Parasoft Load Test.*
- **WebKing Paths:** WebKing's Path view has been replaced by Test Suite-based functional tests using Browser Playback tools. The primary benefit is that Test Suite-based functional tests support much more complex web applications (like RIA and AJAX applications). Moreover, the new implementation follows a consistent test configuration paradigm that supports end-to-end testing. Paths in existing .wkj files can be executed in SOAtest 9.x, but they cannot be edited or extended.
- **WebKing Publishing:** This functionality is not applicable to SOAtest 9.x.
- **Capture HTTP Traffic Tool:** This tool is no longer supported. If this functionality is needed, a free tool like WireShark can be used to save the HTTP trace to a file, then the "Generate tests from traffic" option can be used to create tests from it.
- **Specific XML Validator Options:** The XML DTD preferences and validating against DTD options are no longer available.
- **Management Reports:** Report enhancements are planned. SOAtest will report all meta-data to Report Center and Report Center will be able to generate different kinds of reports.
- **CLI commands:**
    - -run: This command, which is for running custom Jython scripts through SOAtest, is deprecated. Please contact Technical Support for assistance migrating scripts to 9.x and later.
    - -runtest: This command is replaced with new CLI options. See the Migration Guide topic on Command Line Interface Migration for details.
    - -wsdl
    - -reportAllTraffic
    - -traffic