Adding Projects, .tst files, and Test Suites

This topic provides a general guide on how to add projects, .tst files and test suites using SOAtest's various test creation wizards.

Sections include:

Wizard Descriptions

For a description of the various wizards available for adding projects, .tst files, and test suites, see Wizards for Creating Projects, .tst Files, and Tests.

Projects, .tst files, and Test Suites

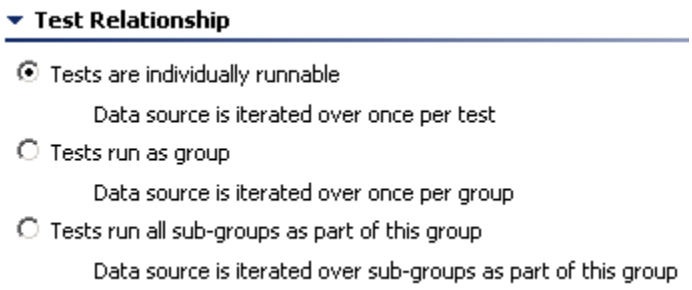
A project (an entity created by Eclipse) can contain any number of SOAtest-specific .tst files. They can also contain source files you want to analyze with SOAtest, and any other resources that make sense for your environment.

Each .tst file can include any number of test suites/scenarios, tools, and inputs. The organization and structure is up to you. To keep file size down and to improve maintainability, we recommend using one .tst file for each distinct testing requirement.

For best practices related to projects, test files, and workspaces, see Workspaces, Projects, and Test Files.

Test Suites and Scenarios

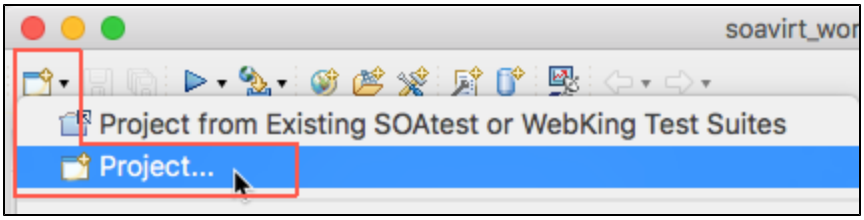
A test suite is any collection of tests that are individually runnable, and has the following setting in the test suite configuration panel:



A scenario is any collection of tests that are not individually runnable because they have dependencies. One example of a scenario is when a series of API tests extracts a value from one test's response and uses it as part of subsequent test message. Another example is a sequence of web scenarios recorded from a browser.

Creating an Empty Project

1. Open the **New** drop-down menu and choose **Project...**

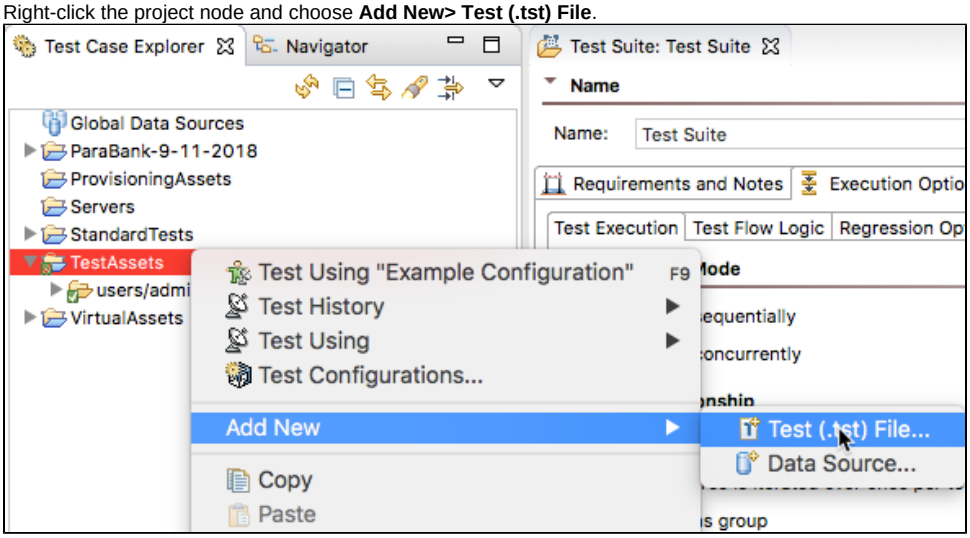


2. Choose **SOAtest> Empty Project**, then click **Next**.
3. Enter a name for the project and change the destination if necessary.
4. Click **Finish**.

Adding a New .tst File to an Existing Project

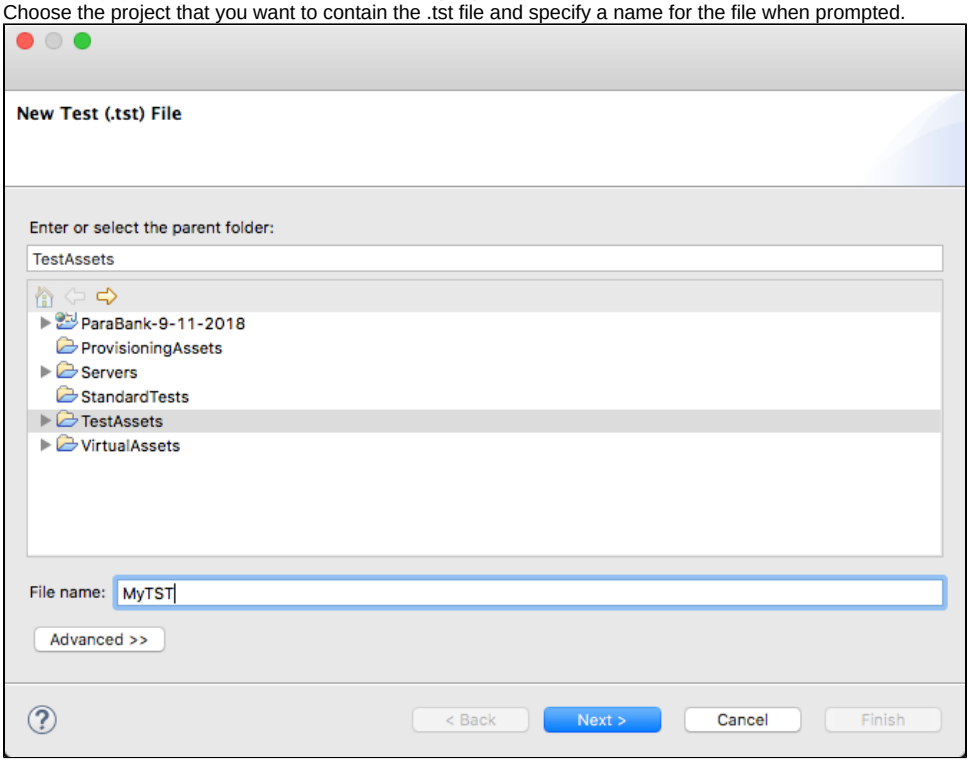
We recommend creating a separate test (.tst file) for each distinct requirement.

1. Right-click the project node and choose **Add New> Test (.tst) File**.



Alternatively, you can choose **File > New > Test (.tst) File** from the main menu.

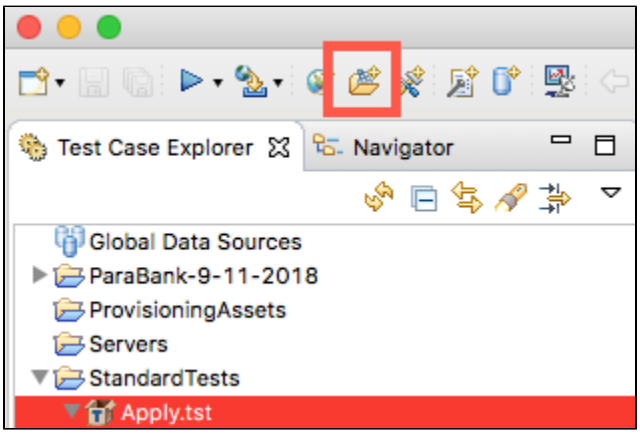
2. Choose the project that you want to contain the .tst file and specify a name for the file when prompted.



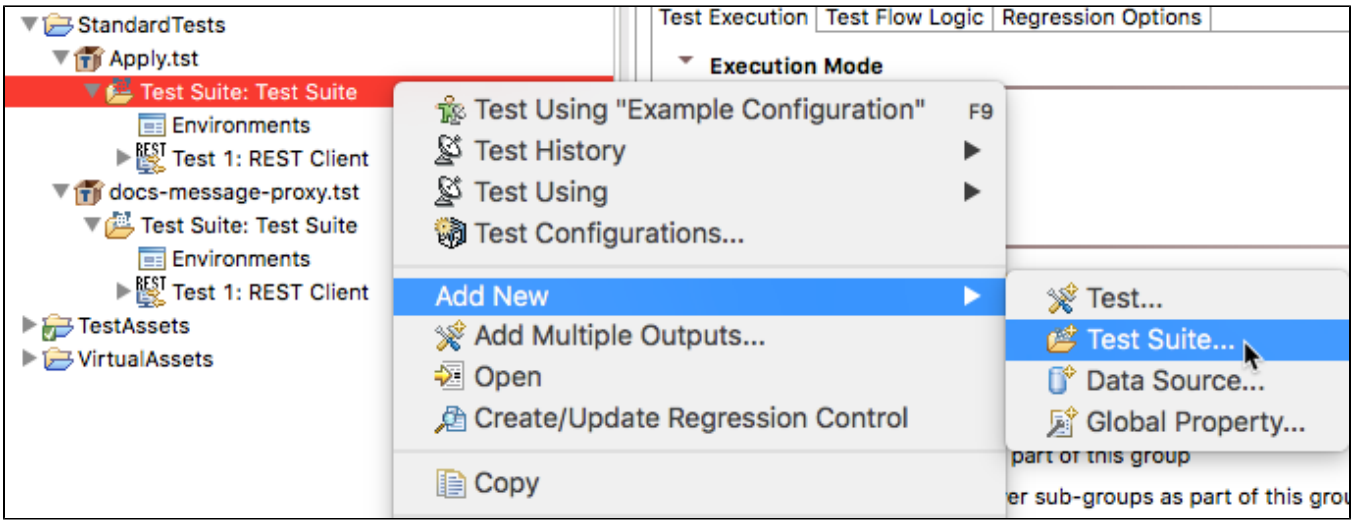
3. Click **Next** and complete the wizard to specify what type of tests you want to create and how you want them created. For help selecting and completing the available test creation wizards, see Wizards for Creating Projects, .tst Files, and Tests.

Adding a New Test Suite

Choose a node in the Test Case Explorer and click the **Add Test Suite** button in the toolbar:



Alternatively, you can right-click a Test Case Explorer node and choose **Add New> Test Suite**.

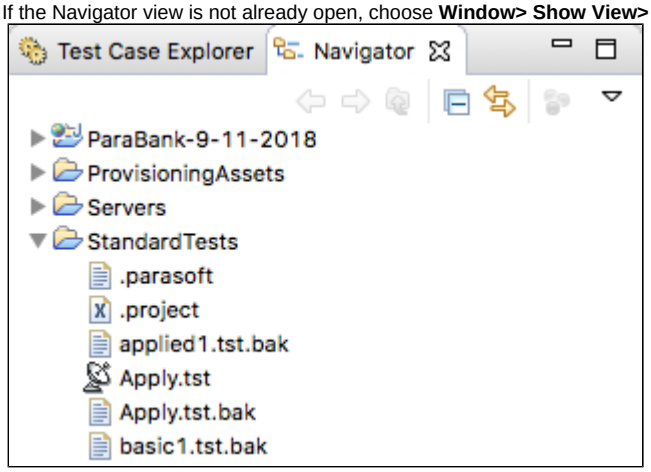


For help selecting and completing the available test creation wizards, see Wizards for Creating Projects, .tst Files, and Tests.

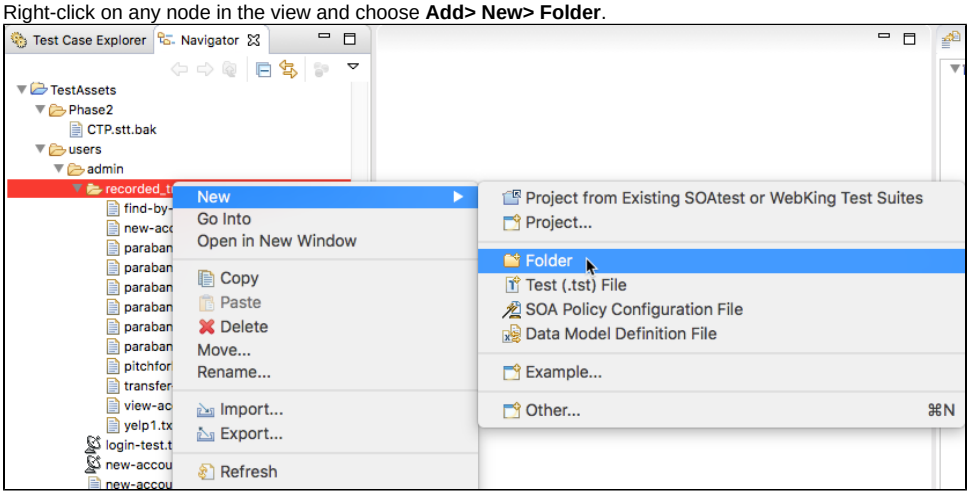
Organizing Project Files

You can create folder structures in your projects to keep your work organized.

1. If the Navigator view is not already open, choose **Window> Show View> Navigator** from the main menu.



2. Right-click on any node in the view and choose **Add> New> Folder**.



3. When prompted, verify that the location for the new folder is correct and specify a name in the Folder name field. You can change the location by clicking on a different project or subfolder. You can also manually add subfolders to the project by specifying a path in the parent folder field.
4. Drag files and folders into the new folder.

Add additional folders as necessary to organize your files.

Using Eclipse Java Projects in SOAtest

Creating a New SOAtest Java Project

SOAtest allows you to create a new Eclipse Java project that has access to SOAtest's Extensibility API, then configure SOAtest scripts and Extension tools to invoke classes from the new Java project.

To create a new SOAtest Java project:

1. Choose **File> New> Project**.
2. Select **SOAtest> Custom Development> SOAtest Java Project**, then click **Next**.
3. Complete this wizard, which has the same options as Eclipse's Java Project wizard.
4. Click **Finish**.

Your new Java project will be shown in the Package Explorer view in the Eclipse Java development perspective. The project's build path will automatically have the jar files needed in order to use SOAtest's Extensibility API. Any Java classes added to your project can be accessed by Extension tools in your SOAtest test suite. For an example of how to do this, see "Java Example" in Extensibility and Scripting Basics.

Using an Existing Java Project

To use an existing Java project from your workspace, you must first add that Java project to SOAtest's classpath as follows:

1. Choose **Parasoft> Preferences**.
2. Open the **Parasoft> System Properties** page.
3. Click the **Add Java Project** button and choose the appropriate project.

The selected Java Project's build output folder and build path entries will be added to the classpath table.

If the **Automatically reload classes** option is enabled, then SOAtest will attempt to reload classes from your Eclipse project after being modified or recompiled. The **Reload** button can also be used to force SOAtest to reload classes from the classpath entries.