

# Running Tests in Different Environments

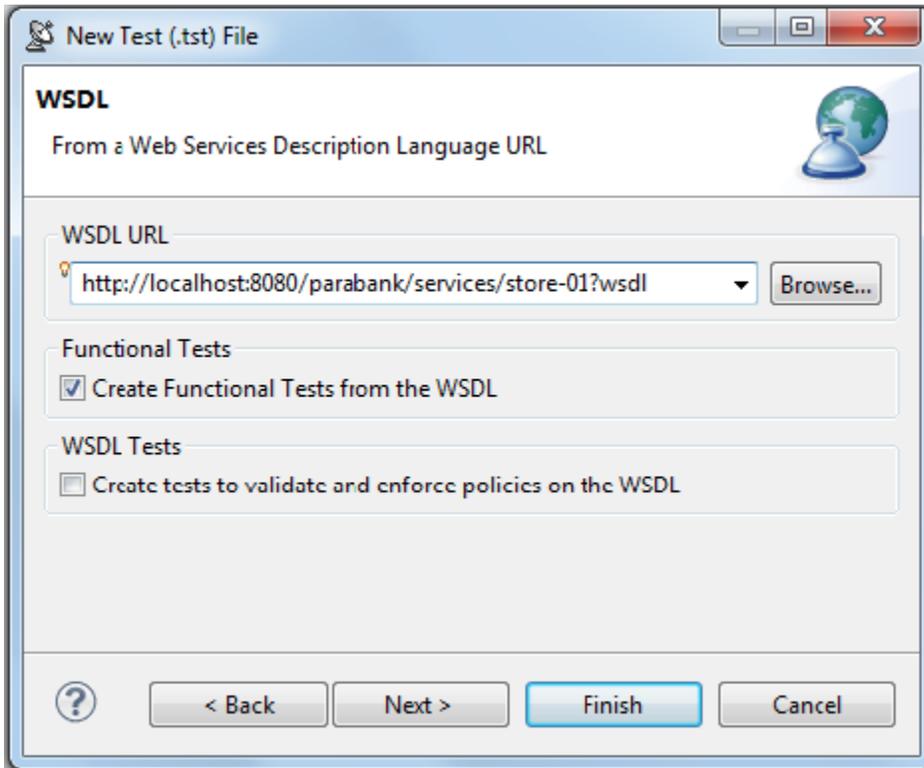
It is common for SOA applications to be developed and maintained by different teams under different environments.

For example, a developer may start with tests on a local server. Then, as the app is deployed to a development build server, the same tests need to be executed against that server. Later, QA and testing teams need to run the same regression tests on their own integration server.

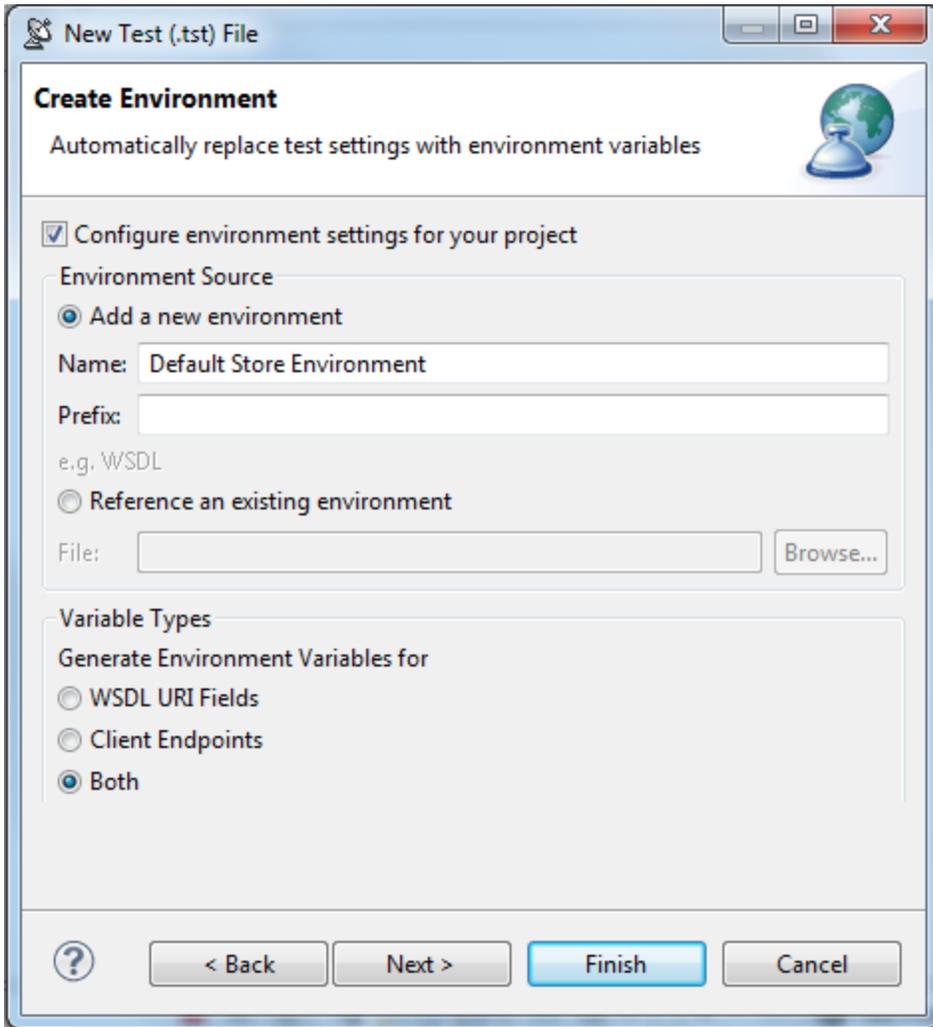
Parasoft SOAtest's environments management feature facilitates such tasks because reusing and sharing test assets is critical for achieving a highly efficient process. One common way is to have SOAtest configure automatically-generated tests for environments, then modify environment settings so that the generated tests can be more easily used in different environments—without requiring any changes to the actual tests.

To create a new test suite with preconfigured environment variables:

1. Right-click the project from the previous exercises, then choose **Add New> Test (.tst) File** from the shortcut menu.
2. Enter a name for the file, then click **Next**.
3. Select **SOA> WSDL** and click **Next**.
4. Enter `http://localhost:8080/parabank/services/store-01?wsdl` in the **WSDL URL** field.
5. Ensure that **Create Functional Tests From the WSDL** is checked.

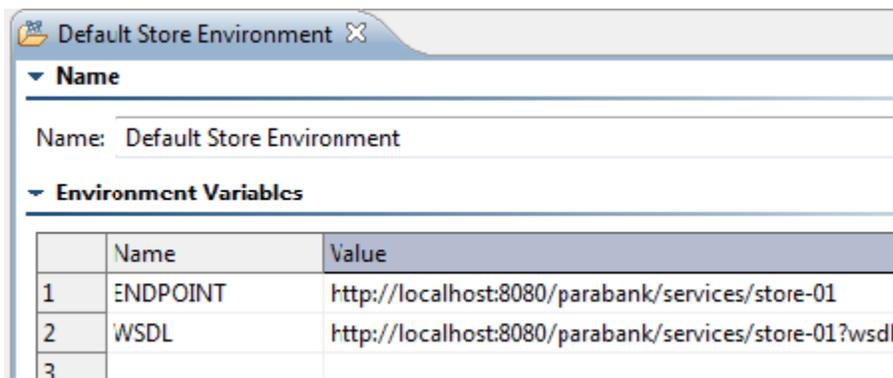


6. Click the **Next** button twice to advance to the Create Environment page.
7. Check **Configure environment settings for your project**, enter `Default Store Environment` under the **Name** field, select **Both** under **Variable Types**, then click **Finish**.

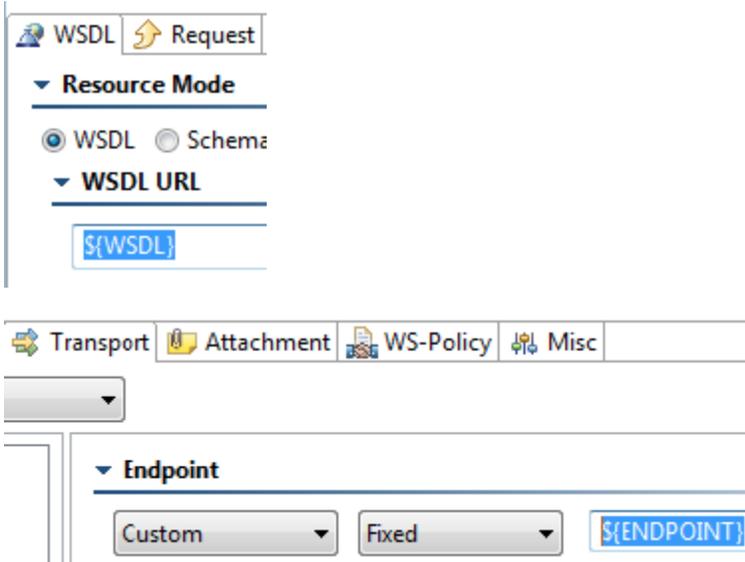


A new **Test Suite: Test Suite** node displays in the Test Case Explorer tab. New environment variables are added to the Default Store Environment node.

8. Double-click the **Default Store Environment** node that was added to the new test suite. Notice how the environment configuration now includes variables for the WSDL and Endpoint.



9. Open one of the generated SOAP Client tests and notice that the same variables shown above are referenced by name. For example:



10. Run the getItemByTitle test. The SOAP messages will be sent to the original store service.



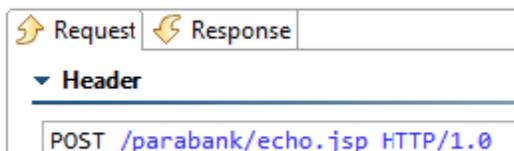
To configure SOAtest to run these same tests in another environment, create a new environment configuration:

1. Right-click the **Environments** node and select **New Environment**.
2. In the New Environment configuration panel, and enter `Echo Environment` in the **Name** field.
3. In the table, enter the following variable names and values:
  - ENDPOINT: `http://localhost:8080/parabank/echo.jsp`
  - WSDL: `http://localhost:8080/parabank/services/store-01?wsdl`

▼ **Environment Variables**

	Name	Value
1	ENDPOINT	<code>http://localhost:8080/parabank/echo.jsp</code>
2	WSDL	<code>http://localhost:8080/parabank/services/store-01?wsdl</code>

4. Click **Save** to save the new environment.
5. Right-click the new **Echo Environment** node and select **Set as Active Environment** from the shortcut menu. This will set that new environment as the new configuration for the test project.
6. Run the tests. The SOAP messages will now be sent to `echo.jsp` instead of the original store service.



Note that:

- Environment *configurations* such as Default Store Environment and Echo Environment can be exported and imported into external XML files, as well as uploaded and referenced to the Parasoft Team Server.
- Environment *variables* such as `$(ENDPOINT)` and `$(WSDL)` can be referenced from most of the fields in the test settings GUI—not just URL fields.
- In the data source editor, use the `soa_env` prefix to reference environment variables. For example, `${soa_env:Variable1}/calc_values.xlsx`

## Environments in Web Scenarios

Environments can also be deployed to assist in web scenario execution—in much the same way that they can be used in service functional testing.

When a web functional test is created, a new environment is created. By default, it contains the base URL for the web site you are testing (named {DOMAIN\_COM\_BASE\_URL}). You can create additional environments that can redirect the test initiation to another base URL (e.g., if you are testing a web application that is deployed locally).

## Switching Environments for Automated Regression Testing

One of the greatest benefit of environments is the ability to rerun the same regression suites from the command line without having to open the SOAtest GUI and modify host or URL settings.

For an example of how to do this, see [Applying an Environment Configuration to a Regression Test from the Command Line](#).