

SilkTest® 2010

Quick Start Tutorial for the SilkTest Recorder

**Borland Software Corporation
4 Hutton Centre Dr., Suite 900
Santa Ana, CA 92707**

Copyright 2009-2010 Micro Focus (IP) Limited. All Rights Reserved. SilkTest contains derivative works of Borland Software Corporation, Copyright 1992-2010 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS and the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

BORLAND, the Borland logo and SilkTest are trademarks or registered trademarks of Borland Software Corporation or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

All other marks are the property of their respective owners.

Contents

- Quick Start Tutorial for the SilkTest Recorder.....4**
- Starting SilkTest Recorder.....4
- Recording a Test Case for the Insurance Company Web Site.....4
- Replaying the Test Case for the Insurance Company Web Site.....5
- Exporting a Test Case or Project.....6
 - Exporting a Project to Silk4J.....6
 - Exporting a Test Method to Silk4J.....6
 - Exporting a Project to SilkTest Classic.....7
 - Exporting a Test Case to SilkTest Classic.....8
 - Exporting a Silk4NET Project.....8
 - Exporting a Silk4NET Test9
- Modifying the Test Case for the Insurance Company Web Site.....10

Quick Start Tutorial for the SilkTest Recorder

This tutorial provides a step-by-step introduction to using the SilkTest Recorder to test a Web site. The Recorder uses *dynamic object recognition* to record and replay test cases that use XPath queries to find and identify objects.

The Recorder can record and replay tests for the following application types:

- Adobe Flex
- Java SWT
- SAP
- Windows API-based client/server
- Windows Forms
- Windows Presentation Foundation (WPF)
- xBrowser (Web applications)

For the sake of simplicity, this guide assumes that you have installed the Recorder and are using the Insurance Company sample Web site, <http://demo.borland.com/InsuranceWebExtJS/>.


For additional information about Recorder, refer to the Recorder Help. To view the help, open Recorder and then choose **Help > Help Contents**.

Starting SilkTest Recorder

Choose **Start > Programs > Silk > SilkTest <version> > SilkTest Recorder**.
Recorder opens and the SilkTest Open Agent icon appears in the system tray.

Recording a Test Case for the Insurance Company Web Site

Record a test case for the Insurance Company sample Web site to see how the Recorder creates a test.

1. Perform one of the following steps:
 - Click  in the toolbar.
 - Choose **Record > Start Recording**.The **New Application Configuration** wizard opens.
2. Double-click **Web Site Test Configuration**.
The **New Web Site Configuration** page opens.
3. From the **Browser Type** list box, select **Internet Explorer**.
You can use Firefox to replay tests but not to record them.
4. Perform one of the following steps:

- **Use existing browser** – Click this option button to use a browser that is already open when you configure the test. For example, if the Web page that you want to test is already displayed in a browser, you might want to use this option.
- **Start new browser** – Click this option button to start a new browser instance when you configure the test. Then, in the **Browse to URL** text box specify the Web page to open.

For this tutorial, close all open browsers and then click **Start new browser** and specify <http://demo.borland.com/InsuranceWebExtJS/>.

5. Click **Finish**.


The Web site opens. The Recorder creates a base state and starts recording.

6. In the Insurance Company Web site, perform the following steps:

- From the **Select a Service or login** list box, select **Auto Quote**.
The **Automobile Instant Quote** page opens.
- Type a zip code and email address in the appropriate text boxes, click an automobile type, and then click **Next**.
- Specify an age, click a gender and driving record type, and then click **Next**.
- Specify a year, make, and model, click the financial info type, and then click **Next**.
A summary of the information you specified appears.
- Point to the **Zip Code** that you specified and press **Ctrl+Alt** to add a verification to the script.
You can add a verification for any of the information that appears.
The **Verify Properties** dialog box opens.
- Check the **textContent** check box and then click **OK**.
A verification action is added to the script for the zip code text.

An action that corresponds with each step is recorded.

7. In the Recorder, perform one of the following steps:

- Click  in the toolbar.
- Choose **Record > Stop Recording**.

8. Choose **File > Save**.

- Navigate to the location in which you want to save the test.
- In the **File name** text box, type the name for the test and then click **Save**.
For example, type **ZipTest**.

Replay the test to ensure that it works as expected. You can modify the test to make changes if necessary.


Replaying the Test Case for the Insurance Company Web Site

Replay a test to ensure that it works as expected.

1. On the main window, from the **Replay speed** list box, select the speed to use to replay the test.

- **Fast** – This option is the fastest choice for test replay and the true speed at which scripts are executed. The other speeds use a delay mechanism, so you can see the test run.
- **Medium** – In most cases, this option enables you to see each action.
- **Slow** – To ensure that you see each action, choose this option.

You can determine the speed of the test by checking the **Replay duration** time posted next the **Replay speed** list box.

2. To replay the entire test, click  in the toolbar or choose **Replay ► Replay All**. The Recorder plays back the test.
3. If an error occurs, perform one of the following steps:
 - Click **Retry** to replay the current action.
 - Click **Stop** to end the test.
 - Click **Skip** to advance to the next action in the test.


Export the test to use it with SilkTest, Silk4NET, or Silk4J. Or, modify the test to make changes if necessary.

Exporting a Test Case or Project

Use the following procedures to export a test case or project to SilkTest Classic, Silk4NET, or to Silk4J.

Exporting a Project to Silk4J

Export projects to use Silk4J as the primary GUI for tests and to organize test methods.

1. Choose **File ► Export**.
The **Export** wizard opens.
2. Double-click **Export as Silk4J Project**.
The **Export as Silk4J Project** page opens.
3. In the **Project location** text box, specify the location to which to export the project.
Optional: Click  and navigate to the folder that you want to use.
4. In the **Project name** text box, specify the project name.
For example, type `web sample Project`.
5. In the **Package** text box, specify the package name.
For example, type `com.example`.
6. In the **Test class** text box, specify the class name to which the test belongs.
For example, type `AutoTests`.
7. In the **Test method** text box, specify a name for the test method.
For example, type `TestAutoInput`.
8. From the **File encoding** list box, select the type of file encoding to use.
9. Click **Finish**.

The Recorder creates a project and exports it to Silk4J.

Import the project with Silk4J. The new project contains a base state and test method and is ready for testing. For details about importing a project, refer to the *Silk4J User Guide*.

Exporting a Test Method to Silk4J

Export test methods to use Silk4J as the primary GUI for tests or to copy the JTF script to the clipboard.

1. Choose **File ► Export**.

The **Export** wizard opens.

2. Double-click **Export as JTF Script**.

The **Export to JTF** page opens.

3. From the **Export to** list box, select one of the following options:

- **Clipboard** – Copies the JTF script to the clipboard. You might choose this option if you want to copy and paste the script into an existing JTF script.
- **JTF Script** – Exports the script to Silk4J. You might choose this option if you want to create a new script or overwrite an existing JTF script.

4. In the **Test method** text box, specify a name for the test method.

For example, type `TestAutoInput`.


5. In the **Package** text box, specify the package name.

For example, type `com.example`.

6. In the **Test class** text box, specify the class name to which the test belongs.

For example, type `AutoTests`.

7. In the **Source folder** text box, specify the location to which to export the test.

Optional: Click  and navigate to the folder that you want to use.

8. From the **File encoding** list box, select the type of file encoding to use.

9. To include the base state in the exported script, check the **Use base state** check box.

The base state makes sure that the application that you want to test is running and in the foreground. This ensures that tests will always start with the same application state, which makes them more reliable. In order to use the base state, it is necessary to specify what the main window looks like and how to launch the application that you want to test if it is not running. Creating a base state is optional. However, it is recommended as a best practice.

If you export to JTF, a separate `silk4j.settings` file is created for the base state. If you export to the clipboard, a `Before` method includes the base state.

10. Click **Finish**.

Recorder creates a Java script and exports it to Silk4J or to the clipboard.

Exporting a Project to SilkTest Classic

Export projects to use SilkTest Classic as the primary GUI, group test cases into projects, or to share data with other products such as SilkCentral Test Manager.


1. Choose **File > Export**.

The **Export** wizard opens.

2. Double-click **Export as SilkTest Project**.

The **Export as SilkTest Project** page opens.

3. In the **Project location** text box, specify the location to which to export the project.


Optional: Click  and navigate to the folder that you want to use.

4. In the **Project name** text box, specify the project name.

For example, type `Web Sample Project`.

5. In the **4Test script** text box, specify the script file name.

For example, type `AutoTests.t`.

Optional: Click  and navigate to the folder that you want to use.


6. In the **Test case** text box, specify a name for the test case.
For example, type `testAutoInput`.
7. To start SilkTest Classic after the test case is exported, check the **Open the exported project in SilkTest** check box.
8. Click **Finish**.

The Recorder creates a project that includes a script that uses the 4Test language and a recovery file and exports the project to SilkTest Classic.

Use SilkTest Classic to work with the exported project. The new project contains a base state and test case and is ready for testing.

Exporting a Test Case to SilkTest Classic

Export test cases to use SilkTest Classic as the primary GUI or to copy the script to the clipboard.


1. Choose **File > Export**.
The **Export** wizard opens.
2. Double-click **Export as 4Test Script**.
The **Export to 4Test Script** page opens.
3. From the **Export to** list box, select one of the following options:
 - **Clipboard** – Copies the script to the clipboard. You might choose this option if you want to copy and paste the script into an existing 4Test script.
 - **4Test Script** – Exports the script to SilkTest Classic. You might choose this option if you want to create a new script or overwrite an existing script.
4. In the **Test case** text box, specify a name for the test case.
For example, type `testAutoInput`.
5. In the **4Test script** text box, specify the script file name.
For example, type `AutoTests.t`.
Optional: Click  and navigate to the folder that you want to use.
6. To start SilkTest Classic after the test case is exported, check the **Open the exported script in SilkTest** check box.
7. Click **Finish**.

The Recorder creates a script that uses the 4Test language and exports it to SilkTest or the clipboard.

Exporting a Silk4NET Project

Export a Silk4NET test and create a new Visual Studio project that uses VB.NET or C# as the primary programming language.

1. Choose **File > Export**.
The **Export** wizard opens.
2. Double-click **Export as Silk4NET Project**.
The **Export as Silk4NET Project** page opens.
3. From the **Programming language** list box, specify whether the project uses VB.NET or C#.
4. In the **Project location** text box, specify the location to which to export the project.

Optional: Click  and navigate to the folder that you want to use.


5. In the **Project name** text box, specify the project name.
For example, type `VB.NET Sample Project`.
6. In the **Namespace** text box, specify the container name for the project.
7. In the **Test class** text box, specify the class name to which the test belongs.
For example, type `AutoTests`.
8. In the **Test method** text box, specify a name for the test method.
For example, type `TestAutoInput`.
9. Click **Finish**.

Recorder creates a project that includes a test that uses the VB.NET or C# language.

Use Visual Studio or a C# editor to work with the exported project. The new project contains a base state and test case and is ready for testing.

Exporting a Silk4NET Test

Export a Silk4NET test that you can add to a Visual Studio project. During the export setup, you can choose to create the test in either VB.NET or C#. Additionally, you can use the clipboard feature to copy and paste the recorded test steps into an existing Silk4NET test.



1. Choose **File > Export**.
The **Export** wizard opens.
2. Double-click **Export as NTF Script**.
The **Export to NTF** page opens.
3. From the **Export to** list box, select one of the following options:
 - **Clipboard** – Copies the recorded test steps to the clipboard. Choose this option if you want to copy and paste the test steps into an existing Silk4NET test.
 - **NTF Script** – Exports the recorded test steps as a Silk4NET test you can add to an existing Visual Studio project. Choose this option if you want to create a new test or overwrite an existing Silk4NET test.
4. From the **Programming language** list box, specify whether the test uses VB.NET or C#.
5. In the **Test method** text box, specify a name for the test method.
For example, type `TestAutoInput`.
6. In the **Namespace** text box, specify the container name for the test.
7. In the **Test class** text box, specify the class name to which the test belongs.
For example, type `AutoTests`.
8. In the **Source folder** text box, specify the location to which to export the test.
Optional: Click  and navigate to the folder that you want to use.
9. To include the base state in the exported script, check the **Use base state** check box.
The base state makes sure that the application that you want to test is running and in the foreground. This ensures that tests will always start with the same application state, which makes them more reliable. In order to use the base state, it is necessary to specify what the main window looks like and how to launch the application that you want to test if it is not running. Creating a base state is optional. However, it is recommended as a best practice.
10. Click **Finish**.

Recorder exports a Silk4NET test or copies the recorded test steps to the clipboard.


Modifying the Test Case for the Insurance Company Web Site

After you record a test case, perform this step to manually modify the test. For example, you might add an additional action, change the order of the recorded steps, or change the parameters for a specific action.

1. Open the test that you want to modify.
2. Navigate to the Web page on which the test ended in the initial recording.
For instance, if the existing test includes multiple Web pages, navigate to the last page in the test.

In this case, the existing test recording ended on the **Automobile Instant Quote** summary page. We want to return to the Home page, which the base state will do automatically when we record.
3. To record the modifications, perform the following steps:
 - a) Click  in the toolbar.
 - b) From the **Select a Service or login** list box, select **Agent Lookup**.
The **Find an Insurance Co. Agent** page opens.
 - c) Click  in the toolbar.
4. To change an existing action, parameter, or locator string, perform the following steps:
 - a) Click the row that you want to change in the Actions grid.
For example, click the row that contains the **SetText** action for the e-mail address parameter.
The **Action Details** tab shows the **Locator**, **Action**, and **Parameters**.
 - b) To change the locator, type a string in the **Locator** text box.
The locator string identifies the object that you want to test.
 - c) Click **Validate Locator**.
Recorder validates the new locator. If the string is not valid or the object cannot be found, an error is displayed.
 - d) To change the action, select an action from the **Action** list.
 - e) To change the parameters, type a parameter value in the appropriate text box.
For example, change the **text** parameter to **"tutorial@yourcompany.com"** for the **SetText** action.
Any changes that you make display in the Actions grid immediately.
5. To manually add an action, perform the following steps:

Typically, you record actions that you want to add to a test. However, you can also manually add an action by copying and pasting an existing action that you then modify.

 - a) Click an action in the **Actions** list.
 **Note:** You can insert multiple actions by pressing **Ctrl** or **Shift** and then clicking the actions that you want to copy.
 - b) Perform one of the following steps:
 - Choose **Edit > Copy Selected Actions** and then choose **Edit > Paste Actions**.
 - Press **Ctrl+C** and then press **Ctrl+V**.A new action displays below the action that you selected.
 - c) In the Action Details view, modify the action to meet your needs by changing the locator, action name, or parameters as needed.
6. Choose **File > Save**.

Replay the test to ensure that it works as expected.

Index

B

base state
 run before record/replay 10

E

exporting
 Silk4NET tests 9
 SilkTest projects 8
 test cases to Silk4J 6
 test cases to SilkTest 7, 8

R

recording
 additional actions 10

recording (*continued*)
 test cases 4
 replaying test cases 5

S

starting Recorder 4

T

test cases
 exporting 6, 7, 8, 9
 modifying 10
 recording 4