Silk Test 18.5

Using the Basic Workflow with the Open Agent

Contents

Testing a Web Application with Dynamic Object Recognition

This tutorial describes how you can use dynamic object recognition in Silk Test Classic to test a Web application. Dynamic object recognition enables you to write test cases that use XPath queries to find and identify objects.

To perform the tasks described in this tutorial, install Silk Test. The examples in the tutorial use the Green Mountain Outpost (GMO) Web application, *http://demo.borland.com/gmopost/*.

For additional information about Silk Test Classic, including information about sample scripts and applications, refer to the Silk Test Classic Help located in (in Microsoft Windows 7) **Start** > **Programs** > **Silk** > **Silk Test** > **Documentation** > **Silk Test Classic** or (in Microsoft Windows 10) **Start** > **Silk**.

Starting Silk Test Classic

Click (in Microsoft Windows 7) **Start** > **Programs** > **Silk** > **Silk Test** > **Clients** > **Silk Test Classic** or (in Microsoft Windows 10) **Start** > **Silk** > **Silk Test Classic**. Silk Test Classic opens.

Creating a New Project

The type of project that you select determines the default Silk Test Agent. For Web application projects, the Open Agent is automatically set as the default agent. Silk Test Classic uses the default agent when configuring an application and recording a test case.

1. In Silk Test Classic, click **File** > **New Project**, or click **Open Project** > **New Project** on the basic workflow bar. the **Create Project** dialog box appears.
2. Type a unique name for the project into the **Project Name** field.

   If the project name already exists, you are prompted to specify a different project name or overwrite the existing project.

   Project files with a `.vtp` (Verify Test Project) extension, `projectname.vtp`, and a `projectname.ini` are created in the directory that you specify in the **Save in** text box.
3. Click **OK**. Silk Test Classic creates a `<Project name>` folder within the specified directory, saves the `projectname.vtp` and `projectname.ini` to this location and copies the extension .ini files, which are `appexpex.ini`, `axext.ini`, `domex.ini`, and `javaex.ini`, to the `extend` subdirectory. If you do not want to save your project in the default location, click **Browse** and specify the folder in which you want to save your project. Silk Test Classic then creates your project and displays nodes on the **Files** and **Global** tabs for the files and resources associated with this project.

Configuring the Insurance Company Web Application

When you configure an application, Silk Test Classic automatically creates a base state for the application. An application's base state is the known, stable state that you expect the application to be in before each test begins execution, and the state the application can be returned to after each test has ended execution.

1. Click **Configure Applications** on the **Basic Workflow** bar.

   If you do not see **Configure Applications** on the **Basic Workflow** bar, ensure that the default Agent is set to the Open Agent.

The **Select Application** dialog box opens.

2. Select the **Web** tab.

3. Select **Internet Explorer**.

4. In the **Enter URL to navigate** text box, type the URL of the web application that you want to test.

   For example, type *http://demo.borland.com/gmoajax*.

5. Click **OK**.

   The **Choose name and folder of the new frame file** page opens. Silk Test Classic configures the recovery system and names the corresponding file `frame.inc` by default.

6. Navigate to the location in which you want to save the frame file.

7. In the **File name** field, type the name for the frame file that contains the default base state and recovery system. Then, click **Save**.

   Silk Test Classic automatically creates a base state for the application. By default, Silk Test Classic lists the caption of the main window of the application as the locator for the base state. When you configure an application, Silk Test Classic adds an include file based on the technology or browser type that you enable to the **Use files location** in the **Runtime Options** dialog box. For instance, if you configure an application that uses one of the supported browsers, Silk Test Classic adds the `xBrowser.inc` file to the **Runtime Options** dialog box.

Silk Test Classic opens the Web page. Record the test case whenever you are ready.

Recording a Test Case for the Insurance Company Web Site

1. Click **Record Testcase** on the **Basic Workflow** bar. The **Record Testcase** dialog box opens.

2. Type the name of your test case in the **Testcase name** field.

   For example, type `ZipTest`.

   Test case names are not case sensitive; they can be any length and consist of any combination of alphabetic characters, numerals, and underscore characters.

3. From the **Application State** list box, select **DefaultBaseState** to have the built-in recovery system restore the default base state before the test case begins executing. If you choose **DefaultBaseState** as the application state, the test case is recorded in the script file as: `testcase testcase_name ()`.

4. If you do not want Silk Test Classic to display the status window during playback when driving the application to the specified base state, uncheck the **Show AppState status window** check box.

   Typically, you check this check box. However, in some circumstances it is necessary to hide the status window. For instance, the status bar might obscure a critical control in the application you are testing.

5. Click **Start Recording**. Silk Test Classic:

   - Closes the **Record Testcase** dialog box.
   - Starts your application, if it is not already running
   - Removes the editor window from the display.
   - Displays the **Recording** window.
   - Waits for you to take further action.

6. In the insurance company web site, perform the following steps:

   a) From the **Select a Service or login** list box, select **Auto Quote**. The **Automobile Instant Quote** page opens.

   b) Type a zip code and email address in the appropriate fields, click an automobile type, and then click **Next**.

   c) Specify an age, click a gender and driving record type, and then click **Next**.

d) Specify a year, make, and model, click the financial info type, and then click **Next**. A summary of the information you specified displays.

e) 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 displays.

The **Verify Properties** dialog box opens.

f) Check the **textContents** 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. To review what you have recorded, click **Stop** in the **Recording** window.

Silk Test Classic displays the **Record Testcase** dialog box, which contains the code that has been recorded for you.

8. Click **Paste to Editor**.

9. Click **File** > **Save**.

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

Replaying a Test Case for the Insurance Company Web Site

Replay a test to ensure that it works as expected.

1. Make sure that the test case you want to run is in the active window.

2. Click **Run Testcase** on the **Basic Workflow** bar.

Silk Test Classic displays the **Run Testcase** dialog box, which lists all the test cases contained in the current script.

3. Select the ZipTest test case.

4. To wait one second after each interaction with the application under test is executed, check the **Animated Run Mode (Slow-Motion)** check box.

Typically, you will only use this check box if you want to watch the test case run. For instance, if you want to demonstrate a test case to someone else, you might want to check this check box. Executions of the default base state and functions that include one of the following strings are not delayed:

- `BaseStateExecutionFinished`
- `Connecting`
- `Verify`
- `Exists`
- `Is`
- `Get`
- `Set`
- `Print`
- `ForceActiveXEnum`
- `Wait`
- `Sleep`

5. Click **Run**.

Silk Test Classic runs the test case and generates a results file. The results file describes whether the test passed or failed, and provides summary information.