

# Silk Test 17.5

Silk Test Classic Quick Start Tutorial  
for Dynamic Object Recognition

**Micro Focus**  
**The Lawn**  
**22-30 Old Bath Road**  
**Newbury, Berkshire RG14 1QN**  
**UK**  
<http://www.microfocus.com>

**Copyright © Micro Focus 2016. All rights reserved.**

**MICRO FOCUS, the Micro Focus logo and Silk Test are trademarks or registered trademarks of Micro Focus IP Development Limited 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.**

**2016-11-02**

# Contents

<b>Testing a Web Application with Dynamic Object Recognition</b> .....	<b>4</b>
Starting Silk Test Classic .....	4
Creating a New Project .....	4
Configuring the GMO Web Application .....	4
Create a Test Case for the GMO Web Application .....	5
Running the Sample Test Case for the GMO Web Application .....	6

# 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/>.

## 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 GMO Web Application

Configure the web application that you want to test to set up the environment that Silk Test Classic will create each time you record or replay a test case. If you are testing a web application or an application that uses a child technology domain of the xBrowser technology domain, for example an Apache Flex application, use this configuration.

1. Click **Configure Applications** on the basic workflow bar.  
If you do not see **Configure Applications** on the 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 the browser that you want to use from the list of available browsers.  
If you want to record a test against a web application, select a browser that is supported for recording. To see which browsers are supported for recording, refer to the [Release Notes](#).
4. In the **Enter URL to navigate** field, type `http://demo.borland.com/gmopost/`.
5. *Optional:* Check the **Create Base State** check box to create a base state for the application under test.  
By default, the **Create Base State** check box is checked for projects where a base state for the application under test is not defined, and unchecked for projects where a base state is defined. 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. When you configure an application and create a base state, 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.
6. Click **OK**.
  - If you have checked the **Create Base State** check box, the **Choose name and folder of the new frame file** page opens. Silk Test Recorder configures the recovery system and names the corresponding file `frame.inc` by default.
  - If you have not checked the **Create Base State** check box, the dialog box closes and you can skip the remaining steps.
7. Navigate to the location in which you want to save the frame file.
8. In the **File name** text box, type the name for the frame file that contains the default base state and recovery system. Then, click **Save**. Silk Test Classic creates a base state for the application. By default, Silk Test Recorder lists the caption of the main window of the application as the locator for the base state. Then Silk Test Classic opens the Web page.
9. Record the test case whenever you are ready.

## Create a Test Case for the GMO Web Application

You must configure the GMO Web application before performing this task.

When you record a test case with the Open Agent, Silk Test Classic creates locator keywords in an INC file to create scripts that use dynamic object recognition and window declarations. This test case will click the **Enter GMO OnLine** button in the GMO sample application.

1. Click **Record Testcase** on the basic workflow bar. If the workflow bar is not visible, click **Workflows > Basic** to enable it. The **Select Test Type** dialog box opens.
2. Select **4Test Testcase**.
3. Click **OK**. The **Record Testcase** dialog box opens.
4. Type `EnterGMOOnLine` in the **Testcase name** text box.
5. 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.
6. Click **Start Recording**. Silk Test Classic performs the following actions:
  - Closes the **Record Testcase** dialog box.
  - Starts your application, if it was not already running. If you have not configured the application yet, the **Select Application** dialog box opens and you can select the application that you want to test.
  - Removes the editor window from the display.
  - Displays the **Recording** window.
  - Waits for you to take further action.
7. In the GMO Web application, click **Enter GMO OnLine**.  
As you interact with the application, Silk Test Classic displays your actions in the **Recorder** window.

8. To stop recording, click **Stop** in the **Recorder**. Silk Test Classic displays the **Record Testcase** dialog box, which contains the code that has been recorded for you.
9. To add the recorded interactions to a script, click **Paste to Editor** in the **Record Testcase** window. If you have interacted with objects in your application that have not been identified in your include files, the **Update Files** dialog box opens.
10. Click **Paste testcase and update window declaration(s)** and then click **OK**. Silk Test Classic creates a new untitled 4Test script in the editor and pastes the actions you have recorded into this script.
11. Choose **File > Save** to save the script. The **Save As** dialog box opens.
12. In the **File name** text box, type `SimpleBrowserTest` and then click **Save**. The **Add to Project** dialog box opens, asking you if you want to add the script file to the project.
13. Click **Yes** Silk Test Classic adds the file to the project scripts and updates the **Files** tree.

The generated test script should look similar to the following:

```
testcase EnterGMOOnline ()  
    recording  
        WebBrowser.BrowserWindow.BSubmit.Click(MB_LEFT, 51, 20)
```

The location of the click may vary, depending on the resolution of your screen and the size of the browser window.

## Running the Sample Test Case for the GMO Web Application

When you run a test case, Silk Test Classic interacts with the application by executing all the actions you specified in the test case and testing whether all the features of the application performed as expected.

Silk Test Classic always saves the suite, script, or test plan before running it if you made any changes to it since the last time you saved it. By default, Silk Test Classic also saves all other open modified files whenever you run a script, suite, or test plan. To prevent this automatic saving of other open modified files, uncheck the **Save Files Before Running** check box in the **General Options** dialog box.

1. Make sure that the test case that you want to run is in the active window.
2. Click **Run Testcase** on the **Basic Workflow** bar.  
If the workflow bar is not visible, choose **Workflows > Basic** to enable it.  
Silk Test Classic displays the **Run Testcase** dialog box, which lists all the test cases contained in the current script.
3. In the **Testcase** list box, click **EnterGMOOnline**.
4. Click **Run**. Silk Test Classic runs the test case and generates a results file.