

SilkTest[®] Classic

Using the Basic Workflow
With the Classic Agent

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.

July 2010

Table of Contents

Overview of the Basic Workflow	4
Creating a SilkTest Project.....	5
Enabling and Testing Extension Settings.....	6
Configuring the Recovery System	7
Recording a Testcase	8
Running a Testcase	10
Exploring Results	11

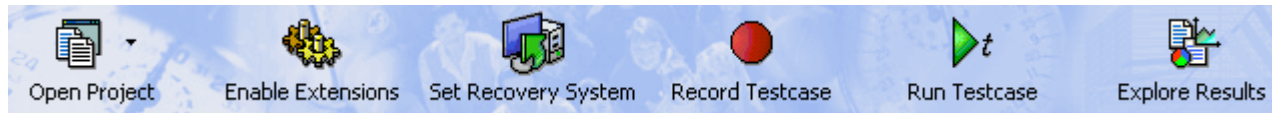
Overview of the Basic Workflow

The Basic Workflow bar guides you through the process of creating a testcase.

To create and execute a testcase, click each icon in the workflow bar to perform the relevant procedures including:

- Create a project
- Enable extensions
- Setting the recovery system
- Record a testcase
- Run a testcase
- View test results using the Silk TrueLog Explorer

This workflow is on by default; you can turn it on and off by choosing **Workflows/Basic**.



Extensions that SilkTest can automatically configure

Use the Basic Workflow to have SilkTest Classic automatically enable extensions for:

- Standalone Java AWT applications
- Java Swing applications
- Java applets
- Powerbuilder applications
- Web applications

For specific information about supported versions, see the Release Notes by choosing **Start/Programs/Silk/SilkTest <version>/Release Notes**.

The Basic Workflow does not automatically configure browser applications containing ActiveX objects. To configure a browser application with ActiveX objects, check the ActiveX check box in the row for the extension you are enabling in the Extensions dialog.

If the Basic workflow does not support your configuration, you can enable the extension manually. See *Enabling extensions manually* in the *online Help* and the *Release Notes* for more information.

Creating a SilkTest Project

You can use the Basic Workflow bar to create a new project or open an existing project. Only one project can be open at a time. In this tutorial, we'll create a new project for use with the GMO Web application. The examples in this tutorial use Internet Explorer 7 and the GMO Web application, available at: <http://demo.borland.com/gmopost/>

- 1 In SilkTest Classic, click **Open Project/New Project** on the Basic workflow bar.
- 2 On the New Project dialog, under Generic, click **Classic Agent** and then click **OK**.
- 3 On the Create Project dialog, type **GMO_Test** in the Project Name field and **create GMO testcase using Basic Workflow bar** in the Description field.
- 4 Click **OK** to save your project in the default location, <SilkTest installation directory>\Project.

When you click OK, SilkTest:

- creates a *projectname* folder, in this case, GMO_Test, within the <SilkTest installation directory>\Projects directory
- writes the project file, GMO_Test.vtp, and GMO_Test.ini to this location
- copies the SilkTest extension .ini files (appexpex.ini, axext.ini, domex.ini, and javaex.ini) to <SilkTest installation directory>\Projects\GMO_Test\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.

- 5 SilkTest Classic creates your project and displays nodes on the Files and Global tabs for the files and resources associated with this project.

For more information about projects, see *Overview of SilkTest Projects* in the *online Help* or choose **Start/Programs/Silk/SilkTest <version>/Documentation/SilkTest Tutorials** and complete the *Working with Projects* tutorial.

You are now ready to complete the next step on the Workflow bar, Enabling Extensions.

Enabling and Testing Extension Settings

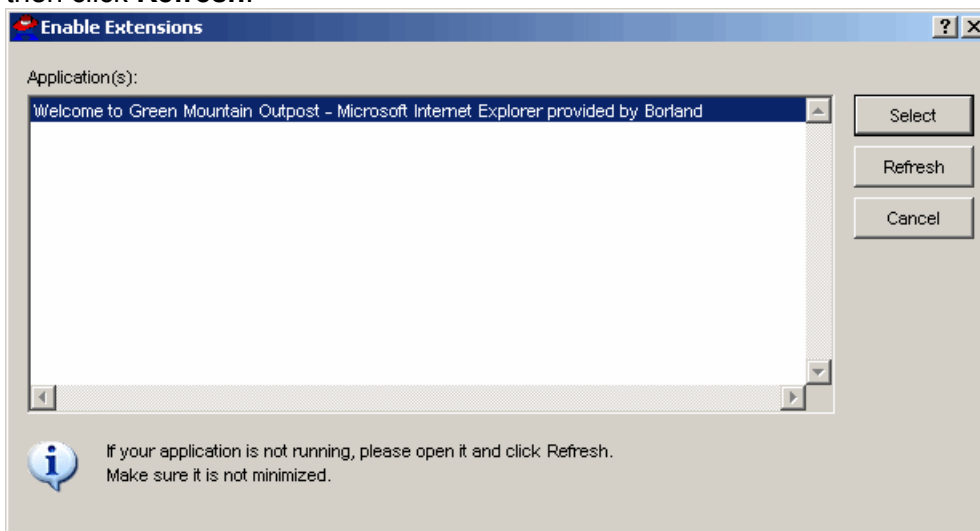
Now that you have created the GMO_Test project, you are ready to use the Basic Workflow bar to enable the appropriate extensions to test the GMO Web application.

An extension is a file that serves to extend the capabilities of, or data available to, a more basic program. SilkTest provides extensions for testing applications that use non-standard controls in specific development and browser environments.

Note

This Web application tutorial uses Internet Explorer 7.0. You can use other supported browsers to perform the procedures in this tutorial. However, slight differences in the window identifiers and other UI elements exist. For details about supported browsers, refer to the *SilkTest Release Notes*.

- 1 Start the application or applet for which you want to enable extensions, in this case: <http://demo.borland.com/gmopost/>. Make sure the application is not minimized.
- 2 Click **Enable Extensions** on the Basic Workflow bar.
- 3 Click **Welcome to Green Mountain Outpost** from the list on the Enable Extensions dialog, and then click **Select**. If the application does not appear in the list, make sure it is not minimized, and then click **Refresh**.



- 4 On the Extension Settings dialog, click **OK** to enable the DOM extension.
- 5 When the Test Extension Settings dialog appears, close the GMO Web application, and then restart it in the same way in which you opened it; for example, by opening Internet Explorer and navigating to <http://demo.borland.com/gmopost/>.
- 6 On the Test Extension Settings dialog, click **Test**.
- 7 Click **OK** on the Test Passed dialog.

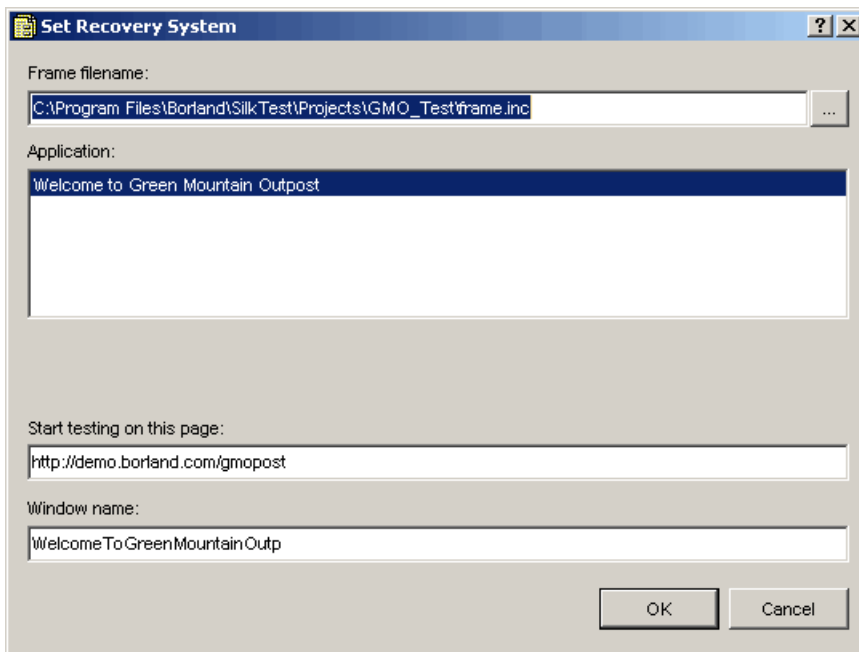
You are now ready to configure the recovery system, which ensures that each testcase begins and ends with the application in its intended state, even if an application error occurs.

Configuring the Recovery System

SilkTest's recovery system allows you to run tests unattended. When your application fails, the recovery system restores the application to a stable state, known as the BaseState, so that the rest of your tests can continue to run unattended.

Using the Basic Workflow bar, you can identify the starting point of the application you are testing, the BaseState, and configure SilkTest's recovery system to return your application to this BaseState: before running a testcase; during a testcase, if an error occurs; and after a testcase completes. For more information about the recovery system, see *Overview of the default recovery system* in the *online Help*.

- 1 Make sure the GMO Web application is running.
- 2 Click **Set Recovery System** on the Basic Workflow bar.
- 3 From the Application list, click **Welcome to Green Mountain Outpost** on the Set Recovery System dialog. Notice that the Frame file name, frame.inc, the URL on which to begin testing, <http://demo.borland.com/gmopost/>, and the window name, WelcomeToGreenMountainOutp, appear.



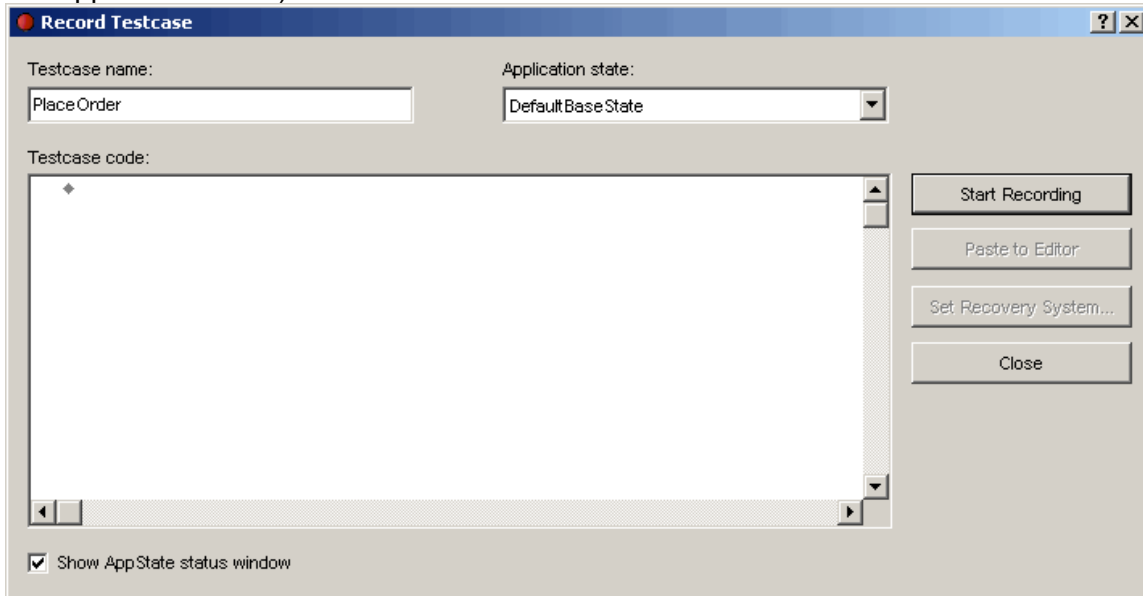
- 4 Click **OK**.
- 5 Click **OK** when the message indicating that the recovery system is configured appears.
- 6 A new 4Test include file, frame.inc, opens in the SilkTest Editor; click the plus sign in the file to see the contents of the frame file.

You are now ready to record a testcase for the GMO Web application.

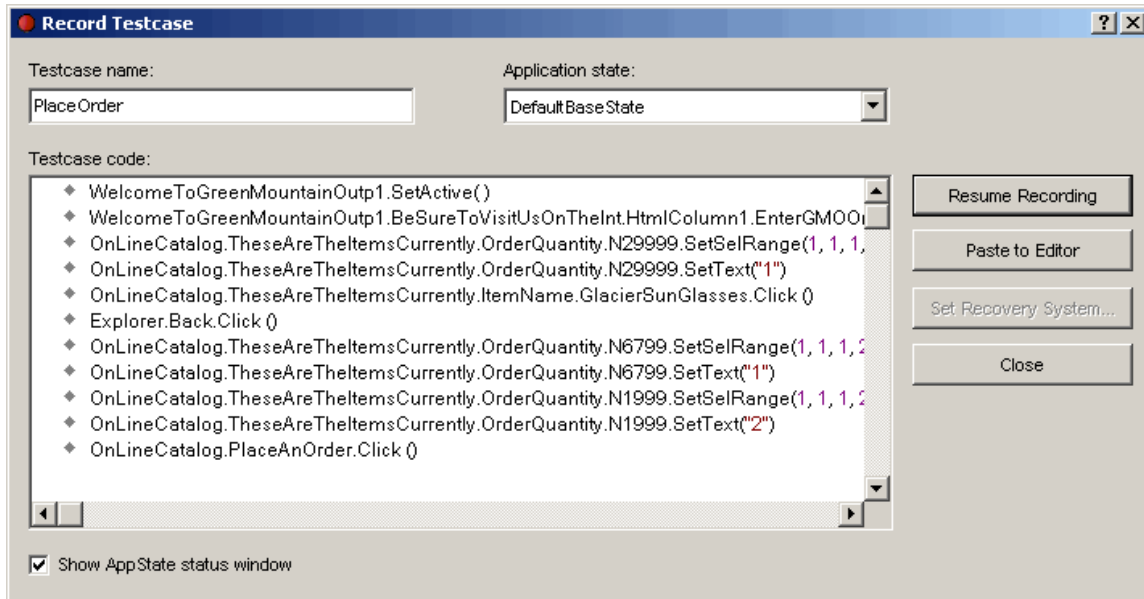
Recording a Testcase

Now that you have configured the recovery system, you are ready to record a testcase. A testcase is an automated test that addresses one objective of a testplan

- 1 On the Basic Workflow bar, click **Record Testcase**.
- 2 On the Record Testcase dialog, type **PlaceOrder** in the Testcase name field. (Since you configured DefaultBaseState when setting up the recovery system, you do not need to modify the Application state.)



- 3 Click **Start Recording**. SilkTest:
 - Closes the Record Testcase dialog
 - Starts your application, if it was not already running
 - Removes the editor window from the display
 - Displays the Record Status window
 - Waits for you to take further action
- 4 When Recording flashes on the Record Status window, begin recording your testcase by clicking the **Enter GMO OnLine** button on the GMO home page. You may notice a rectangle flashing on the OnLine Catalog page as you move your cursor over objects. This is SilkTest's way of letting you know that it has recognized the objects.
- 5 In the Order Quantity column, type **1** in the first row to order a 3 Person Dome Tent.
- 6 Click the **Glacier Sun Glasses** link. Review the description, and then click the **Back** button in the browser. If you make any typos or extra clicks while recording, you can always remove them later.
- 7 Type **1** in the third row of the Order Quantity column to order 1 pair of sunglasses, type **2** in the next row to order two pairs of padded socks, and then click **Place An Order**.
- 8 Click **Done** on the Record Status dialog. The actions you recorded appear in the Record Testcase dialog. If you captured additional cursor movements or clicks the text may be different.

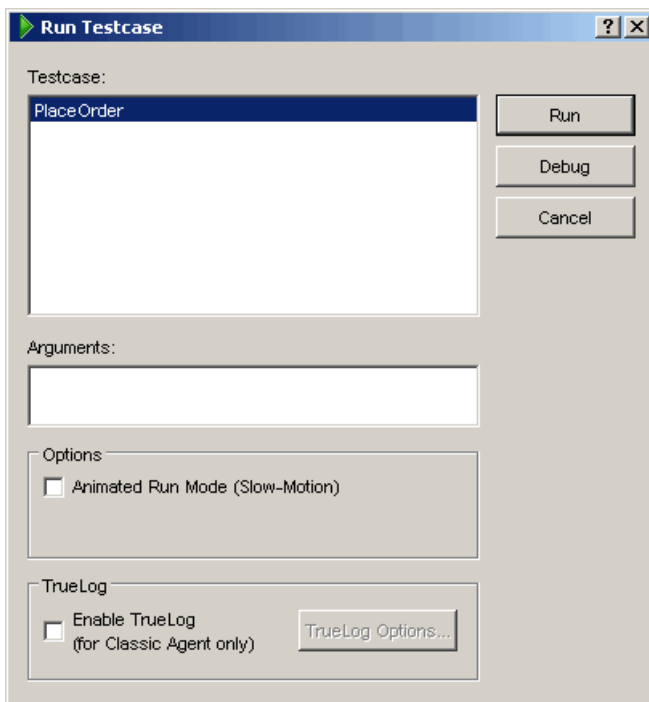


- 9 On the Record Testcase dialog, click **Paste to Editor**.
 - 10 On the Update Files dialog, click **OK** to paste the testcase to the Editor and update the window declarations in the frame file SilkTest created when setting up the recovery system.
 - 11 Click the plus sign in the Editor to see the testcase you created.
 - 12 Choose **File/Save**.
 - 13 In the Save As dialog, specify the file name and location for the testcase and then click **Save**.
 - 14 Click **Yes** to add this file in your project.
- You are now ready to run your testcase.

Running a Testcase

When you run a testcase, SilkTest interacts with the application by executing all the actions you specified in the testcase and testing whether all the features of the application performed as expected.

- 1 Make sure that the testcase you want to run is in the active window.
- 2 On the Basic Workflow bar, click **Run Testcase**.
- 3 SilkTest displays the Run Testcase dialog, which lists all the testcases contained in the current script.



- 4 On the Run Testcase dialog, click **PlaceOrder**, and then click **Run**. SilkTest executes the test on your machine – you can see it replicating the clicks and cursor movements that you recorded. It then displays the results; the PlaceOrder testcase passed.

You can now explore the results.

Exploring Results

Whenever you run tests, SilkTest generates a results file, which indicates how many tests passed and how many failed, describes why tests failed, and provides summary information. You can invoke comparison tools from within the results file that pinpoint exactly how the runtime results differ from your known baselines. Testplan results files offer additional features, such as the ability to generate a Pass/Fail report or compare different runs of the testplan.

SilkTest also provides the Silk TrueLog Explorer to help you analyze test results files. You must configure SilkTest to use the Silk TrueLog Explorer. You can enable TrueLog using the Run Testcase dialog or the TrueLog Options dialog. Using the Run Testcase dialog enables you to enable or disable TrueLog each time you run a testcase. Choose **Options/TrueLog** to open the TrueLog Options dialog and specify what you want to capture. For more information about using Silk TrueLog, see the *Silk TrueLog Explorer for SilkTest User Guide* by choosing **Start/Programs/Silk/SilkTest <version>/Documentation/SilkTest Classic/Silk TrueLog Explorer User Guide**.

- 1 On the Basic Workflow bar, click **Explore Results**.
- 2 On the Results Files dialog, navigate to the file name that you want to review and click **Open**.

By default, the results file has the same name as the executed script, suite, or testplan. To review a file in the SilkTest TrueLog Explorer, open a .xlg file. To review a SilkTest results file in SilkTest, open a .res file.

You have completed the Basic Workflow tutorial and successfully created a project, enabled and tested extension settings, configured the recovery system, and recorded and ran a testcase. For more information about working with SilkTest, see the *online Help*, or access additional tutorials by choosing **Start/Programs/Silk/SilkTest <version>/Documentation/SilkTest Tutorials**.