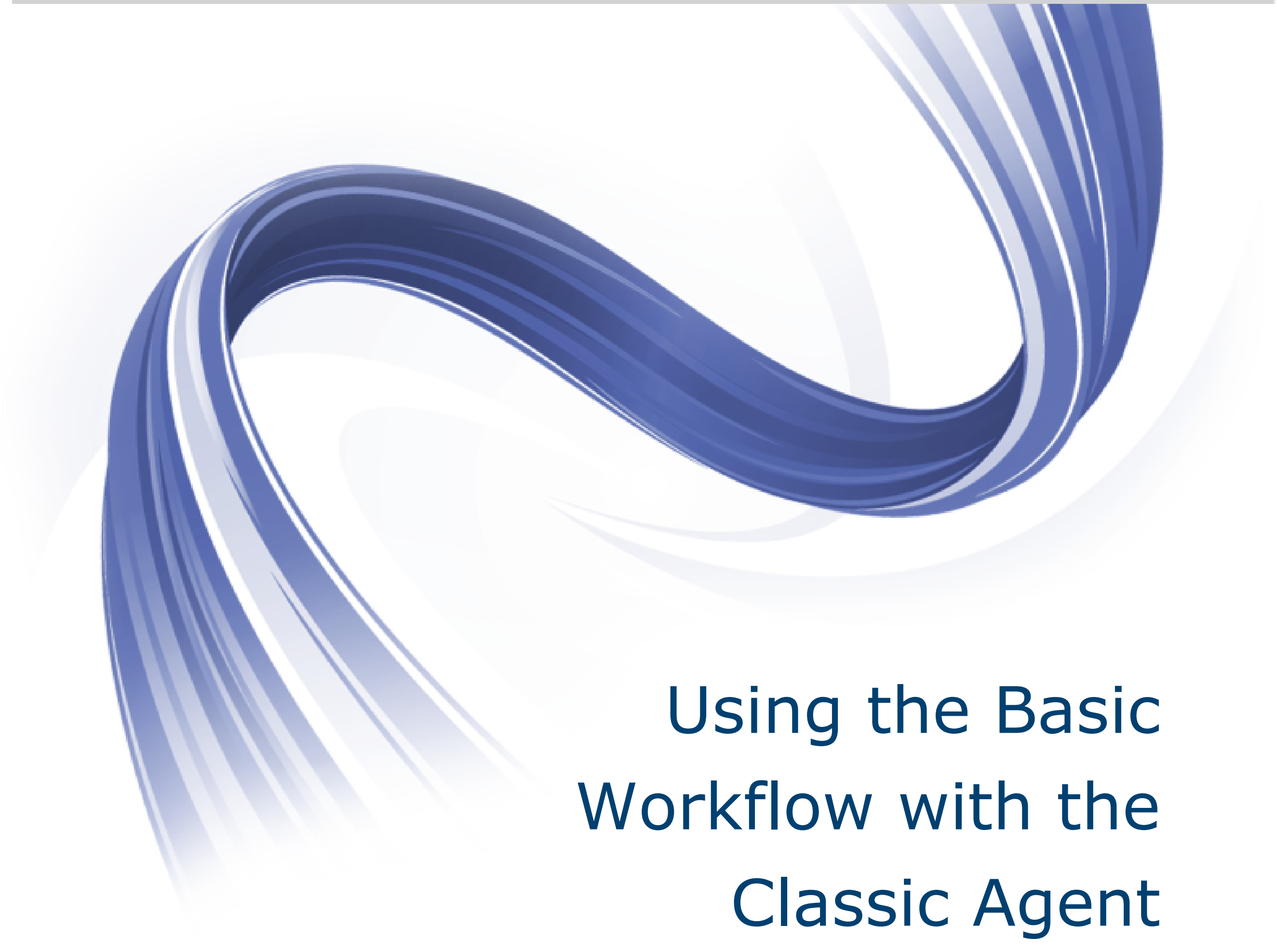
Borland®
(A MICRO FOCUS COMPANY)

MICRO FOCUS®

SilkTest 13.0



Using the Basic Workflow with the Classic Agent

Micro Focus
575 Anton Blvd., Suite 510
Costa Mesa, CA 92626

2012-05-03

Contents

Overview of the Basic Workflow

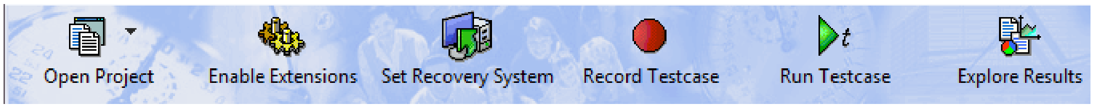
This functionality is available only for projects or scripts that use the Classic Agent.

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

To create and execute a test case, click each icon in the Basic Workflow bar to perform the relevant procedures including:

- Creating a project
- Enabling extensions
- Setting the recovery system
- Recording a test case
- Running a test case
- Viewing test results using Silk TrueLog Explorer

The Basic Workflow is displayed by default; you can display or hide it by clicking **Workflows** > **Basic**.



Extensions that SilkTest Classic can Automatically Configure

This functionality is available only for projects or scripts that use the Classic Agent.

Using the **Basic Workflow**, SilkTest Classic can automatically configure extensions for many development environments, including:

- Browser applications and applets running in one of the following browsers:
  - Windows Internet Explorer
  - Netscape Navigator
  - Mozilla Firefox
  - AOL
- .NET standalone Windows Forms applications
- Windows Presentation Foundation (WPF) applications
- Standalone Java and Java AWT applications
- Java Web Start applications and InstallAnywhere applications and applets
- Java SWT applications
- Visual Basic applications
- Client/Server applications
- Adobe Flex applications

You cannot enable extensions for SilkTest Classic (`partner.exe`), Classic Agent (`agent.exe`), or Open Agent (`openAgent.exe`).

For specific information about supported versions, refer to the *Release Notes*.

If the **Basic Workflow** does not support your configuration, you can enable the extension manually.

If you use the Classic Agent, 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 box. Or, test browser applications using the Open Agent.

Creating a SilkTest Classic Project using the Basic Workflow

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 will 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 from *http://demo.borland.com/gmopost/*.

For additional information about projects, refer to the *SilkTest Classic Help* or click **Start** > **Programs** > **Silk** > **SilkTest** > **Documentation** > **SilkTest Classic** > **Tutorials** and complete the *Working with Projects* tutorial.

1. On the Basic Workflow bar, click **Open Project** > **New Project**. The **New Project** dialog box opens.
2. Under **Generic**, click **Classic Agent**.
3. Click **OK**. The **Create Project** dialog box opens.
4. In the **Project Name** field, type `GMO_Test`.
5. In the **Description** field, type `Create GMO test case using the Basic Workflow bar`.
6. Click **OK**. SilkTest Classic saves the new project in the default location, `<SilkTest Classic installation directory>\Projects`. 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. SilkTest Classic writes the project file, `GMO_Test.vtp`, and `GMO_Test.ini` to this location. SilkTest Classic also copies the SilkTest Classic extension .ini files, which are `appexpex.ini`, `axext.ini`, `domex.ini`, and `javaex.ini`, to `<SilkTest Classic installation directory>\Projects\GMO_Test\extend`.

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

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 the data available to, a more basic program. SilkTest Classic provides extensions for testing applications that use non-standard controls in specific development and browser environments.

Note: This tutorial uses Windows 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 *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. On the Basic Workflow bar, click **Enable Extensions**. The **Enable Extensions** dialog box opens.
3. Click **Welcome to Green Mountain Outpost - Windows Internet Explorer** in the **Application(s)** list box, and then click **Select**. If the application does not appear in the list, make sure it is not minimized, and then click **Refresh**. The **Extension Settings** dialog box opens.
4. Click **OK** to enable the DOM extension.
5. When the **Test Extension Settings** dialog box opens, close and restart the GMO Web application in the same way in which you opened it.

For example, by opening Windows Internet Explorer and navigating to *http://demo.borland.com/gmopost/*.

6. On the **Test Extension Settings** dialog box, click **Test**.

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

Configuring the Recovery System

The recovery system of SilkTest Classic enables 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 the recovery system to return your application to this BaseState in the following cases:

- Before running a test case
- During a test case
- If an error occurs
- After a test case completes

For additional information about the recovery system, refer to the *SilkTest Classic Help*.

1. Make sure the GMO Web application is running.
2. On the Basic Workflow bar, click **Set Recovery System**. The **Set Recovery System** dialog box opens.
3. In the **Application** list box, click **Welcome to Green Mountain Outpost - Windows Internet Explorer**. 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**. A message box displays, stating that the recovery system is configured.
5. Click **OK**. A new 4Test include file, `frame.inc`, opens in the SilkTest Classic Editor.
6. Click the plus sign in the file to see the contents of the frame file.

Recording a Test Case Using the Basic Workflow

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

1. On the Basic Workflow bar, click **Record Testcase**. The **Record Testcase** dialog box opens.
2. In the **Testcase name** field, type `PlaceOrder`.

   Since you configured DefaultBaseState when setting up the recovery system, you do not need to modify the Application state.
3. Click **Start Recording**.

   SilkTest Classic performs the following tasks:

   - Closes the **Record Testcase** dialog box.
   - 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 test case by clicking **Enter GMO OnLine** 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 the way in which SilkTest Classic informs you 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 box. The actions you have recorded display in the **Record Testcase** dialog box.
9. Click **Paste to Editor**. The **Update Files** dialog box opens.
10. Click **OK**. The test case is pasted to the Editor and the window declarations in the frame file are updated.
11. Click the plus sign in the Editor to see the test case that you have created.
12. Click **File** > **Save**. A **Save As** dialog box opens.
13. Specify the **File name** and location for the test case and then click **Save**.
14. Click **Yes** to add this file in your project.

Running a Test Case Using the Basic Workflow

When you run a test case, SilkTest Classic executes all the actions you have specified in the test case and tests whether all the features of the application perform as expected.

1. Make sure that the test case you want to run is in the active window.
2. On the Basic Workflow bar, click **Run Testcase**. The **Run Testcase** dialog box opens, listing all the test cases that are contained in the current script.
3. In the **Testcase** list box, click **PlaceOrder**, and then click **Run**. SilkTest Classic executes the test on your machine. You can see SilkTest Classic replicating the clicks and cursor movements that you have recorded. SilkTest Classic then displays the results.

You can now explore the results.

Exploring Results

Whenever you run tests, SilkTest Classic 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. Test-plan result files offer additional features, such as the ability to generate a Pass/Fail report or compare different runs of the test plan.

You can use Silk TrueLog Explorer to analyze test-result files. Click **Options** > **TrueLog** to open the **TrueLog Options** dialog box and specify what you want to capture. You can also enable Silk TrueLog Explorer in the **Run Testcase** dialog box each time you run a test case.

For additional information about using Silk TrueLog Explorer, refer to the Silk TrueLog Explorer User Guide at **Start** > **Programs** > **Silk** > **SilkTest** > **Documentation** > **SilkTest Classic**.

1. On the Basic Workflow bar, click **Explore Results**. The **Results Files** dialog box opens.
2. 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 test plan. To review a file in the Silk 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 test case. For additional information about working with SilkTest Classic, refer to the *SilkTest Classic Help*. Click **Start** > **Programs** > **Silk** > **SilkTest** > **Documentation** > **SilkTest Classic** > **Tutorials** to access additional tutorials.

Index