



Silk Test 20.0

Using the Basic Workflow with the
Open Agent

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

© Copyright 1992-2019 Micro Focus or one of its affiliates.

MICRO FOCUS, the Micro Focus logo and Silk Test are trademarks or registered trademarks of Micro Focus or one of its affiliates.

All other marks are the property of their respective owners.

2019-04-30

Contents

Basic Workflow for the Open Agent	4
Creating a New Project	4
Configuring Applications	5
Recording Test Cases for Standard and Web Applications	5
Recording Test Cases for Mobile Applications	6
Running a Test Case	7
Viewing Test Results	8

Basic Workflow for the Open 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 workflow bar to perform the relevant procedures. The procedures and the appearance of the workflow bar differ depending on whether your test uses the Open Agent or the Classic Agent.

The **Basic Workflow** bar is displayed by default. You can display it or hide it by checking and un-checking the **Workflows > Basic** check box. If your test uses both the Open Agent and the Classic Agent, the **Basic Workflow** bar changes when you switch between the agents.

When you use the Open Agent, the **Basic Workflow** uses dynamic object recognition to record and replay test cases that use XPath queries to find and identify objects.

Creating a New Project

You can create a new project and add the appropriate files to the project, or you can have Silk Test Classic automatically create a new project from an existing file.

Since each project is a unique testing environment, by default new projects do not contain any settings, such as extensions, class mappings, or Agent options. If you want to retain the settings from your current test set, save them as an options set by opening Silk Test Classic and clicking **Options > Save New Options Set**. You can add the options set to your project.

To create a new project:

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. *Optional:* Type a description for the project into the **Description** field.
After you create the project, you can view the project description by right-clicking the `projectname` folder on the **Files** or **Global** tabs or by clicking **Project > Project Description**.
4. Type the location at which you want to save the project into the **Save in** field.
You can also click **Browse** and specify the folder in which you want to save your project. The default location is `C:\Users\\Documents\Silk Test Classic Projects`.
5. 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.
6. Perform one of the following steps:
 - If your test uses the Open Agent, configure the application to set up the test environment.
 - If your test uses the Classic Agent, enable the appropriate extensions to test your application.



Note: This Help covers the functionality of Silk Test Classic when using the Open Agent. If you are using the Classic Agent, refer to the [Silk Test Classic Classic Agent Help](#).

Configuring Applications

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.

Silk Test Classic has slightly different procedures depending on which type of application you are configuring:

- A standard application, which is an application that does not use a web browser, for example a Windows application or a Java SWT application.
- A web application, which is an application that uses a web browser, for example a web page, a web-based application on a mobile device, or an Apache Flex application.
- A mobile native application, which is a non web-based application on a mobile device.

Recording Test Cases for Standard and Web Applications



Note: Before you can record a test case, you have to configure the application that you want to test.

This functionality is supported only if you are using the Open Agent.

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. With this approach, you combine the advantages of INC files with the advantages of dynamic object recognition. For example, scripts can use window names in the same manner as traditional, Silk Test Classic tag-based scripts and leverage the power of XPath queries.

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 the name of your test case into the **Testcase name** field.
Test case names are not case sensitive; they can have any length and they can consist of any combination of alphabetic characters, numerals, and underscore characters.
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.
 - If you choose *DefaultBaseState* as the application state, the test case is recorded in the script file as `testcase testcase_name ()`.
 - If you choose another application state, the test case is recorded as `testcase testcase_name () appstate appstate_name`.
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 application under test, perform the actions that you want to test.

For information about the actions available during recording, see *Actions Available During Recording*.

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 resume recording your interactions, click **Resume Recording**.
10. 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.
11. Perform one of the following steps:
 - Click **Paste testcase and update window declaration(s)** and then click **OK**. In most cases, you want to choose this option.
 - Choose **Paste testcase only** and then click **OK**. This option does not update the window declarations in the INC file when it pastes the script to the Editor. If you previously recorded the window declarations related to this test case, choose this option.
12. Click **OK**. Silk Test Classic adds the recorded actions to the specified script file, and opens the file in the editor.

Recording Test Cases for Mobile Applications

This functionality is supported only if you are using the Open Agent.

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. With this approach, you combine the advantages of INC files with the advantages of dynamic object recognition. For example, scripts can use window names in the same manner as traditional, Silk Test Classic tag-based scripts and leverage the power of XPath queries.

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 the name of your test case into the **Testcase name** field.

Test case names are not case sensitive; they can have any length and they can consist of any combination of alphabetic characters, numerals, and underscore characters.
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.
 - If you choose *DefaultBaseState* as the application state, the test case is recorded in the script file as `testcase testcase_name ()`.
 - If you choose another application state, the test case is recorded as `testcase testcase_name () appstate appstate_name`.
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. On the device or the Emulator, interact with your native mobile application, driving it to the state that you want to test.
8. In the **Recording** window, perform the actions that you want to record.

- a) Click on the object with which you want to interact. Silk Test Classic performs the default action for the object. If there is no default action, or if you have to insert text or specify parameters, the **Choose Action** dialog box opens.
- b) *Optional:* To choose an action for an object, which might not be the default action, right-click on the object. The **Choose Action** dialog box opens.
- c) *Optional:* If the action has parameters, type the parameters into the parameter fields. Silk Test Classic automatically validates the parameters.
- d) Click **OK** to close the **Choose Action** dialog box. Silk Test Classic adds the action to the recorded actions and replays it on the mobile device or emulator.

For information about how to record an interaction with a mobile device, see *Interacting with a Mobile Device*.

9. To verify an image or a property of a control during recording, click **Ctrl+Alt**.

10. *Optional:* To interact with an object that is currently not visible in the **Recording** window, use the **Hierarchy View**:

- a) Click **Toggle Hierarchy View**. The **Hierarchy View** opens.
- b) In the object tree, right-click on the object on which you want to perform an action.
- c) Click **Add New Action**. The **Choose Action** dialog box opens.
- d) Proceed as with any other action.

For example, to open the main menu of the device or emulator, right-click on the *MobileDevice* object in the object tree and select the action `PressMenu()`.

11. To pause the recording of interactions with the application, for example to move the application into a different state, click **Pause**.

12. To resume recording interactions, click **Start**.

13. To add the recorded interactions to a script, click **Stop**. If you have interacted with objects in your application that have not been identified in your include files, the **Update Files** dialog box opens.

14. Perform one of the following steps:

- Click **Paste testcase and update window declaration(s)** and then click **OK**. In most cases, you want to choose this option.
- Choose **Paste testcase only** and then click **OK**. This option does not update the window declarations in the INC file when it pastes the script to the Editor. If you previously recorded the window declarations related to this test case, choose this option.

Running a Test Case

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. Select a test case and specify arguments, if necessary, in the **Arguments** field.

Remember to separate multiple arguments with commas.

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. To view results using the TrueLog Explorer, check the **Enable TrueLog** check box. Click **TrueLog Options** to set the options you want to record.

6. Click **Run**. Silk Test Classic runs the test case and generates a results file.

For the Classic Agent, multiple tags are supported. If you are running test cases using other agents, you can run scripts that use declarations with multiple tags. To do this, check the **Disable Multiple Tag Feature** check box in the **Agent Options** dialog box on the **Compatibility** tab. When you turn off multiple-tag support, 4Test discards all segments of a multiple tag except the first one.

7. *Optional:* If necessary, you can press both **Shift** keys at the same time to stop the execution of the test.

Viewing Test Results

Whenever you run tests, a results file is generated which indicates how many tests passed and how many failed, describes why tests failed, and provides summary information.

1. Click **Explore Results** on the **Basic Workflow** or the **Data Driven Workflow** bars.

2. On the **Results Files** dialog box, navigate to the file that you want to review and click **Open**.

Silk Test Classic provides the following result files:

- `.res` files, which include information about the execution of the test case, script, suite, test plan, or keyword-driven test.
- `.tlz` files, which include a TrueLog with optional screenshots before and after each action.
- `.htm` for HTML reports with summary information about the test run and detailed information about the executed actions.

By default, Silk Test Classic writes both a TrueLog and an HTML report when running a test. You can select which result formats Silk Test Classic should generate under **Options > TrueLog**.

By default, a results file has the same name as the executed script, suite, or test plan.