

Silk Test 15.5

Silk Test Classic: Working with Projects

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

Copyright © Micro Focus 2014. All rights reserved. Portions Copyright © 1992-2009 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, the Micro Focus logo, and Micro Focus product names 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.

BORLAND, the Borland logo, and Borland product names 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.

2014-06-06

Contents

Silk Test Classic Projects	4
Storing Project Information	4
Accessing Files Within Your Project	5
Sharing a Project Among a Group	5
Creating a New Project	6
Adding Existing Files to a Project	6
Adding New Files to a Project	7
Converting Existing Tests to a Project	7

Silk Test Classic Projects

Silk Test Classic projects organize all the resources associated with a test set and present them visually in the **Project Explorer**, making it easy for you to see your test environment, and to manage it and work within it.

Silk Test Classic projects store relevant information about your project, including the following:

- References to all the resources associated with a test set, such as plans, scripts, data, options sets, .ini files, results, frame files, and include files.
- Configuration information.
- Editor settings.
- Data files for attributes and queries.

All of this information is stored at the project level, meaning that once you add the appropriate files to your project and configure it once, you may never need to do it again. Switching among projects is easy - since you need to configure the project only once, you can simply open the project and run your tests.

When you create a new project, Silk Test Classic automatically uses the agent that is selected in the toolbar.

Each project is a unique testing environment

By default, new projects do not contain any settings, such as enabled extensions, class mappings, or agent options. If you want to retain the settings from your current test set, save them as a options set by opening Silk Test Classic and clicking **Options > Save New Options Set**. You can include the options set when you create your project. You can create a project manually or have automatically generate a project for you, based on existing files that you specify.

Storing Project Information

Silk Test Classic stores project-related information in the following project files:

<code>projectname.vtp</code>	The project file has a Verify Test Project (.vtp) extension and is organized as an .ini file. It stores the names and locations of files used by the project.
<code>projectname.ini</code>	The project initialization file, similar to the <code>partner.ini</code> file, stores information about options sets, queries, and other resources included in your project.
<code>silkTestClassic.ini</code>	A user-specific initialization file that stores user-specific information about the location of the last projects, the size of the project history, and the location of the current project.

These files are created in the `projectname` folder. When you create your project, Silk Test Classic prompts you to store your project in the default location `C:\Users\<<Current user>\Documents\Silk Test Classic Projects`. Silk Test Classic creates a `<Project name>` folder within this directory, saves the `projectname.vtp` and `projectname.ini` to this location and copies the extension .ini files, which are `appexex.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.

When you export a project, the default location is the project directory.



Note: The extension .ini files, which are `appexpex.ini`, `axext.ini`, `domex.ini`, and `javaex.ini`, located in your `<Silk Test Classic installation directory>\extend` folder are copied to the `extend` directory of your project, regardless of what extension you have enabled. Do not rename the `extend` directory; this directory must exist in order for Silk Test Classic to open your project.

You can have Silk Test Classic automatically enable the appropriate extension using the basic workflow bar, or you can manually enable extensions. The current project uses the extension options in the extension .ini file copied to the `extend` directory of your project. Any modifications you make to the options for this enabled extension will be saved to the copy stored within the current project in the `extend` directory.

The `extend` directory is used only for local testing on the host machine. If you want to test on remote agent machines, you must copy the .ini files from the `extend` directory of your project to the `extend` directory on the target machines.

File references

Whether you are emailing, packaging, or adding files to a project, it is important to understand how Silk Test Classic stores the path of the file. The .vtp files of Silk Test Classic use relative paths for files on the same root drive and absolute paths for files with different root drives. The use of relative and absolute file paths is not configurable and cannot be overridden. If you modify the .vtp file to change file references from relative paths to absolute paths, the next time you open and close the project it will have relative paths and your changes will be lost.

Accessing Files Within Your Project

Working with Silk Test Classic projects makes it easy to access your files - once you have added a file to your project, you can open it by double-clicking it in the **Project Explorer**. The **Project Explorer** contains the following two tabs:

Tab	Description
-----	-------------

Files	Lists all of the files included in the project. From the Files tab, you can view, edit, add, and remove files from the project, as well as right-click to access menu options for each of the file types. From the Files tab, you can also add, rename, remove and work with folders within each category.
--------------	--

Global	Displays all the resources that are defined at a global level within the project's files. For example test cases, functions, classes, window declarations, and others. When you double-click an object on the Global tab, the file in which the object is defined opens and your cursor displays at the beginning of the line in which the object is defined. You can run and debug test cases and application states from the Global tab. You can also sort the elements that display within the folders on the Global tab.
---------------	---

Existing test sets do not display in the **Project Explorer** by default; you must convert them into projects.

Sharing a Project Among a Group

Apply the following guidelines to share a Silk Test Classic project among a group:

- Create the project in the location from which it will be shared. For example, you can create the project on a network drive.
- Ensure that testers create the same directory structure on their machines.

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.
2. On the **Create Project** dialog box, type the **Project Name** and **Description**.
3. Click **OK** to save your project in the default location, `C:\Users\<<Current user>\Documents\Silk Test Classic Projects`.

To save your project in a different location, click **Browse** and specify the folder in which you want to save your project.

Silk Test Classic creates a `<Project name>` folder within this 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.

4. 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.

Adding Existing Files to a Project

You can add existing files to a project or create new files to add to the project. We recommend adding all referenced files to your project so that you can easily see and access the files, and the objects contained within them. Referenced files do not have to be included in the project. Plans and scripts will continue to run, provided the paths that are referenced are accurate.

When you add a file to a project, project files (`.vtp` files) use relative paths for files on the same root drive and absolute paths for files with different root drives. The use of relative and absolute files is not configurable and cannot be overridden.

To add an existing file to a project:

1. If your project is not already open, click **File > Open Project** or click **Open Project > Open Project** on the basic workflow bar, select the project to which you want to add a file, and then click **Open**.
2. On the **Project Explorer**, select the **Files** tab, right-click the node associated with the type of file you want to add, and then click **Add File**.

For example, to add a script file to the project, right-click **Script**, and then click **Add File**.

3. On the **Add File to Project** dialog box, specify the file you want to add to the open project, and then click **Open**.

The file name, followed by the path, displays under the appropriate category on the **Files** tab sorted alphabetically by name and is associated with the project through the `projectname.vtp` file. If files exist on a network drive, they are referenced using Universal Naming Conventions (UNC).

You can also add existing files to the project by clicking **Project > Add File**. Silk Test Classic automatically places the file in the appropriate node, based on the file type; for example if you add a file with a .pln extension, it will display under the **Plan** node on the **Files** tab. We do not recommend adding application .ini files or Silk Test Classic .ini files, which are `qaplans.ini`, `propset.ini`, and the `extension.ini` files, to your project. If you add object files, which are .to and .ino files, to your project, the files will display under the **Data** node on the **Files** tab. Objects defined in object files will not display in the **Global** tab. You cannot modify object files within the Silk Test Classic editor because object files are binary. To modify an object file, open the source file, which is a .t or .inc file, edit it, and then recompile.

Adding New Files to a Project

You can create new files to add to a project. We recommend adding all referenced files to your project so that you can easily see and access the files, and the objects contained within them. Referenced files do not have to be included in the project. Plans and scripts will continue to run, provided the paths that are referenced are accurate.

When you add a file to a project, project files (.vtp files) use relative paths for files on the same root drive and absolute paths for files with different root drives. The use of relative and absolute files is not configurable and cannot be overridden.

To add a new file to a project:

1. If your project is not already open, click **File > Open Project** or click **Open Project > Open Project** on the basic workflow bar, select the project to which you want to add a file, and then click **Open**.
2. On the **Project Explorer**, select the **Files** tab, right-click the node associated with the type of file you want to add, and then click **New File**.

For example, to create a script file, right-click **Script**, and then click **New File**.

The new file opens in the 4Test Editor.

3. Save the file and click **Yes** when Silk Test Classic asks you if you want to add the file to the open project.

The file name, followed by the path, displays under the appropriate category on the **Files** tab and is associated with the project through the `projectname.vtp` file. The files in each category are sorted alphabetically by name. If files exist on a network drive, they are referenced using Universal Naming Conventions (UNC).

If you chose to create a new test frame file, the .inc file is automatically added to the project; Silk Test Classic does not prompt you to save it to the project.

You can also create a new file through the **File** menu; for example, to create a new 4Test script, click **File > New > 4Test Script**.

Converting Existing Tests to a Project

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 clicking **Options > Save New Options Set**. You can include the options set when you create your project.

To convert existing test sets to a project:

1. Create a new project.
2. Manually add the files to the project.

Index

A

- accessing
 - files in projects 5
- adding files
 - projects 6
- adding new files
 - projects 7

E

- existing tests
 - adding to projects 7

F

- files
 - adding new files to projects 7
 - adding to projects 6

G

- groups

- sharing projects 5

P

- project-related information
 - storing 4
- projects
 - about 4
 - accessing files 5
 - adding existing tests 7
 - adding files 6
 - adding new files 7
 - creating 6
 - sharing among a group 5
 - storing information 4