Borland
A MICRO FOCUS COMPANY

Silk Test 13.5

Silk Test Classic:
Working with
Projects

Contents

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 supports the type of application that you are testing. For instance, if you create an Adobe Flex project, Silk Test Classic uses the Open Agent.

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 ha 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 two project files:

`projectname.vtp` 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.

`projectname.ini` The project initialization file, similar to the `partner.ini` file, stores information about options sets, queries, and other resources included in your project.

These files are created in the `projectname` folder. When you create your project, Silk Test Classic prompts you to store your project in `<Silk Test Classic installation directory>\Project,` the default location. You can specify another location if you do not want to use the default project folder. Silk Test Classic creates a `projectname` folder within this directory, saves the `projectname.vtp` and `projectname.ini` files to this location, and copies the extension .ini files, which are `appexpex.ini,` `axext.ini, domex.ini,` and `javaex.ini,` to the `projectname\extend` subdirectory.

If you are using Microsoft Windows Vista, project data is saved in the `C:\Users\<username>\AppData` `\Local\VirtualStore\Program Files\Silk\Silk Test <version>` folder, and not in the `Program Files` folder where Silk Test Classic is installed. When you open a project from Silk Test Classic, project data is displayed as expected. However, if you use the File Explorer to access the project, you need to click the **Compatibility Files** icon to access the project. To store project data in the installation folder, as with previous Windows versions, you can disable the Vista User Access Control.

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 your `projectname\extend` directory, regardless of what extension you have enabled. Do not rename your `projectname\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 `projectname\extend` directory. Any modifications you make to the options for this enabled extension will be saved to the copy stored within the current project in your `projectname\extend` directory.

The `projectname\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 `projectname\extend` directory to the `<Silk Test Classic installation directory>\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 **New Project** dialog box, click the type of project that you want to test.

   The type of project that you select determines the default Agent. For instance, if you specify that you want to create an Adobe Flex project, the Open Agent is automatically set as the default agent. Silk Test Classic uses the default agent when recording a test case, enabling extensions, or configuring an application.
3. Click **OK**.
4. On the **Create Project** dialog box, type the **Project Name** and **Description**.
5. Click **OK** to save your project in the default location, `<Silk Test Classic installation directory>\Project`.

   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 `projectname` folder within this directory, saves the `projectname.vtp` and `projectname.ini` files to this location and copies the extension .ini files, which are `appexpex.ini`, `axext.ini`, `domex.ini`, and `javaex.ini` to a `projectname\extend` subdirectory. 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.

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. Choose **File** > **New Project** or click **Open Project** > **New Project** on the basic workflow bar.
2. On the **New Project** dialog box, click **AutoGenerate**.
3. On the **AutoGenerate Project** dialog box, type the **Project Name** and **Description**.
4. Specify the location where you want to save your project.

    We recommend the default location, `<Silk Test Classic installation directory>` `\Project`.

    Silk Test Classic will create a `projectname` folder within this directory, save the `projectname.vtp` and `projectname.ini` to this location and copy the extension .ini files, which are `appexpex.ini`, `axext.ini, domex.ini`, and `javaex.ini`, to a `projectname\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. Click **Add** to select the file from which you want Silk Test Classic to generate a project, for example, a master plan file, and then click **Open**. The file is added to the list in the **Generate from** list box.
6. *Optional:* Click **Add** again to select additional files, if applicable.
7. *Optional:* To add an options set, click **Add**, select **All Files** in the **Files of type** list box, select the options set you want to add, and then click **Open**.
8. *Optional:* To remove a file from the list, select the file, and then click **Remove**.
9. Click **OK** when you have finished adding files. Silk Test Classic automatically generates a new project from the file or files you selected. This may take a few moments, depending on the size and number of files you are converting into projects. Silk Test Classic creates the files `projectname.vtp` and `projectname.ini` in the location that you have you specified in the **Save in** text box and displays the files and resources included in the automatically-generated project in the **Project Explorer**.
10. If you have added an options set and want to use it, you must load it into memory: on the **Files** tab, expand the **Profile** node, right-click the options set you want to load, and then click **Open Options Set**.

    > **Note:** Double-clicking an options set opens it for editing, but does not load it into memory.

Results files, data files, bitmaps, and files included or referenced through file functions are not automatically added to your project. You must add these files manually. You can also convert existing tests to a project by creating a new project and manually adding the files to the project.

Index

A

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

E

existing tests
    converting into 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
    converting existing tests 7
    creating 6
    sharing among a group 5
    storing information 4