Selenium Open Source Test Automation Framework Extensibility for Developers

Version 1.0

September 2009

Table of Contents

1. Purpose of the Document

The purpose of this document is to share guidelines for customizing Open Source Test Automation Framework code. This document will help with adding or modifying functions or keywords in the framework code.

1.1. Scope

The scope of this document is to provide guidelines for customizing Open Source Test Automation Framework code.

1.2. Overview

This document provides guidelines for:

- Adding new keywords
- Modifying functions
- Modifying keywords
- Adding new objects
- Adding new actions

2. Adding New Keywords

New keywords can be added to the framework by following these steps:

- Design the keyword.
- Use “;” as a delimiter in the third column.
- Use “:” as a delimiter in the fourth column.
- Include the second column cell value as a case in the functionlibrary.rb
- Call any existing function or a new function in the functionlibrary.rb
- Use the values of the third and fourth columns to pass parameters to the function and to handle the value that is returned by the function (depending on the function definition).

Example:

To add a keyword for comparing two strings:

1. Design the keyword syntax.

strcompare	<String1>;<String2>	Variable<holds return value>

2. Create a function called “strcompare” in functionlibrary.rb
3. Write the code logic for the function “strcompare” in functionlibrary.rb
4. Create a case statement for “strcompare” in functionlibrary.rb
5. The return value should be true if the two strings match and otherwise should be false.

3. Modifying Keywords

Refer to the Open Source Test Automation Framework Keyword Naming conventions document for a list of keywords that are available in the Open Source Test Automation Framework. The keyword functionality can be modified or customized by following the below steps:

- Identify the keywords that need to be modified.
- Identify where these keywords are used in the framework code and modify them accordingly.

4. Adding New Objects and Actions

4.1. Adding New Objects

Refer to the Open Source Test Automation Framework Keyword Naming Conventions document for a list of objects that are handled in the framework code. When you encounter new or custom objects that are not handled in the framework, you can include them in the framework code by following the below steps:

- Identify the object class and name it based on object naming conventions.
- Identify the list of actions that are performed on the object.
- Identify the list of check points that are needed.
- Include code for setting context on the object.
- Include code for performing actions on the object.
- Include code for performing checks on the object.

4.2. Adding New Actions

When you encounter new actions that are not handled in the framework you can add them into the framework code by following the below steps:

- Identify the action and name it.
- Identify the list of objects where this action should be performed.
- Include code for performing this action for all the required objects.

Example:

Adding a new action: “Rightclick”

1. Place the below code to the perform function in functionlibrary.rb. This will perform the required action on the required object.

   “When "Rightclick"

   @selenium.Rightclick @CurObject”

5. General Guidelines

- Follow coding standards for defining and naming functions.
- Refer to the Open Source Test Automation Framework Keyword Naming Conventions document for including new objects or new actions in the framework code.
- Before adding or modifying functions, add necessary comments such as date of modification, modified by or created by, as well as what was modified or added.
- Use but don't change global variables within the function.
- Use local variables as much as possible.
- Use pass by value and pass by variable for passing values and variables.
- Add new keywords or functions only when existing functions are not able to perform the desired action.