**Test Automation Framework configuration**

**User Manual**

**for**

**--Application--**

This document is being submitted to **Elanco CRDB** by **WinWire Technologies, Inc**. This document and any supporting appendices submitted with it are confidential and intended solely for the use of the business users and IT team working at The Joint Commission.

**Change Record**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | | | **Author** | | **Version** | | **Change reference** |
| DD-MM-YY |  | | | 0.1 | |  | |
| DD-MM-YY |  | | | 0.2 | |  | |
| DD-MM-YY | |  | | 0.3 | |  | |
|  |  | | |  | |  | |
|  |  | | |  | |  | |

**Reviewers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Version approved** | | **Position** | **Date** | |
|  |  |  | | |  |
|  |  |  | | |  |

Table of Contents

[**1.** **Overview** 4](#_Toc509910720)

[**2.** **Prerequisite** 5](#_Toc509910721)

[**3.** **AppName\_TestData.xlsx** 5](#_Toc509910722)

[**4.** **Modules** 9](#_Toc509910723)

[**5.** **App.Config** 10](#_Toc509910724)

[**6.** **Selenium Action (SeleniumActions.cs)** 10](#_Toc509910725)

[**7.** **Hybrid Logic (HybridLogic.cs)** 10](#_Toc509910726)

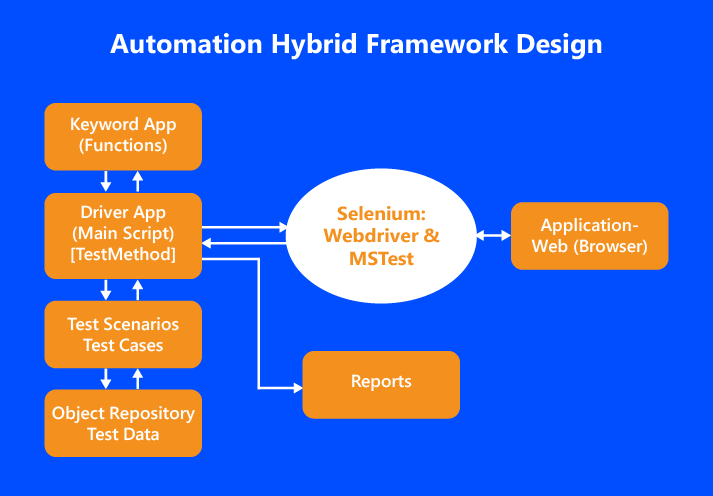
[**8.** **Reports** 11](#_Toc509910727)

[**9.** **XML File** 11](#_Toc509910728)

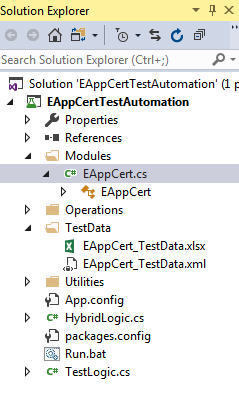
Test Automation Framework configuration user manual

# **Overview**

This document provides information about configuration of test cases for application based on browsers used and modules to be tested. Below is the overview design diagram



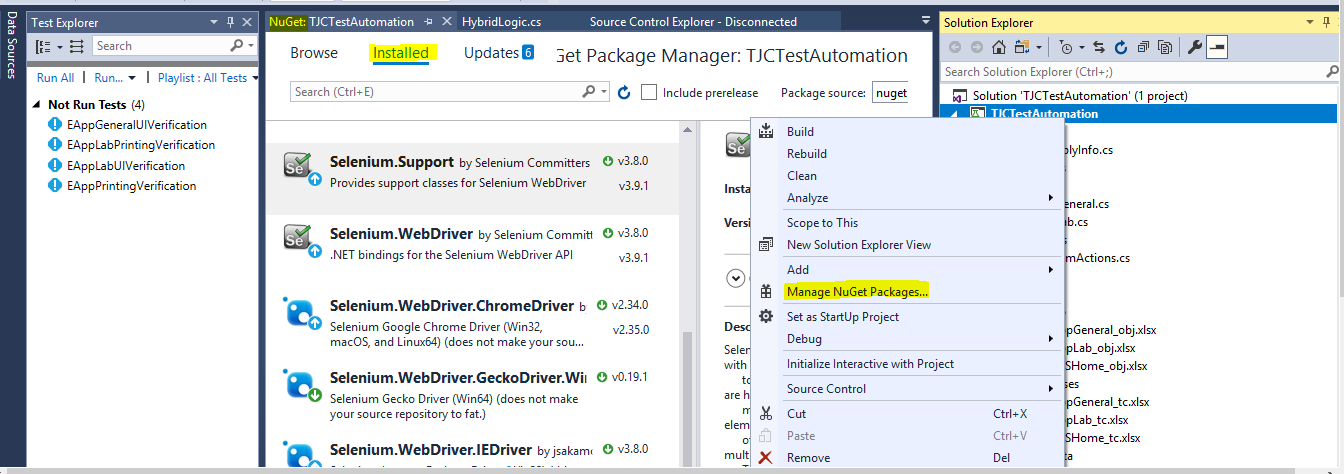
Below is the structure and steps, to configure test modules/ cases for any given application.



# **Prerequisite**

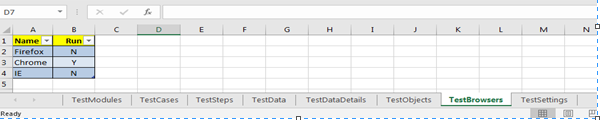
Below are the minimum required NuGet packages to be included in the project

* 1. Selenium.WebDriver
  2. Selenium.Support
  3. Selenium.WebDriver.ChromeDriver
  4. Selenium.WebDriver.GeckoDriver
  5. Selenium.Firefox.WebDriver
  6. Selenium.WebDriver.IEDriver
  7. MSTest.TestAdapter
  8. MSTest.TestFramework
  9. IKVM



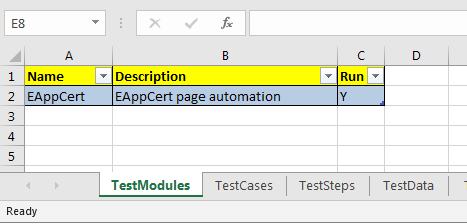
# **AppName\_TestData.xlsx**

**a. TestBrowsers:**



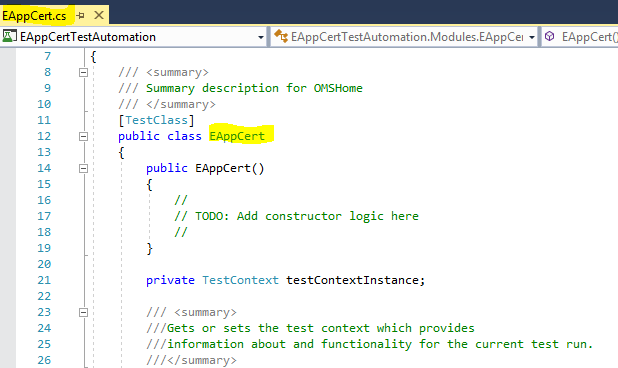
User can enable/ disable any browser by setting value to **‘N’** or **‘Y’** under **‘Run’** column. When the test is executed only the browsers with Y will be run.

**b. TestModules**

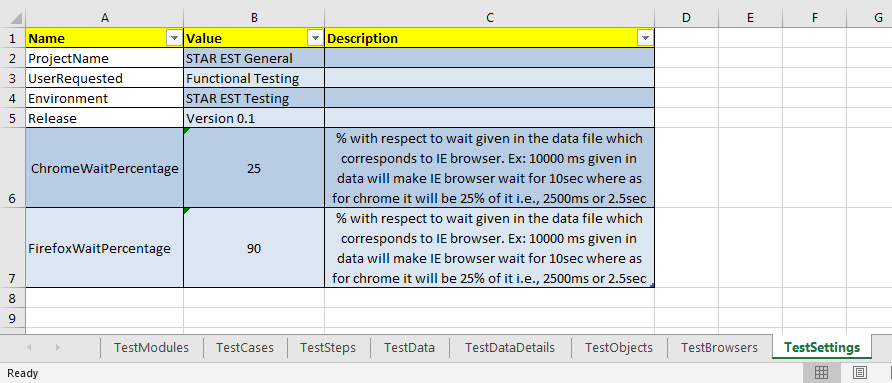


1. User can enable/ disable any **Module** by setting value to **‘N’** or **‘Y’** under **‘Run’** column.

2. User can add any number of Test modules, by entering **‘Name’** value which corresponds to **‘TestClass’** name in the code



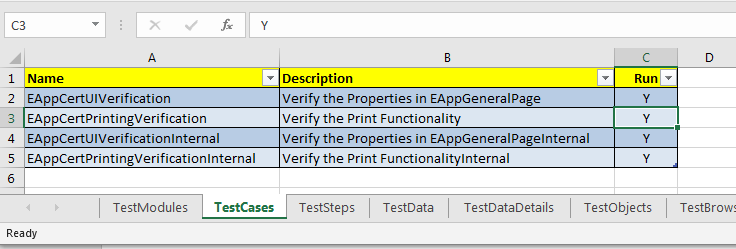
**c. TestSettings**



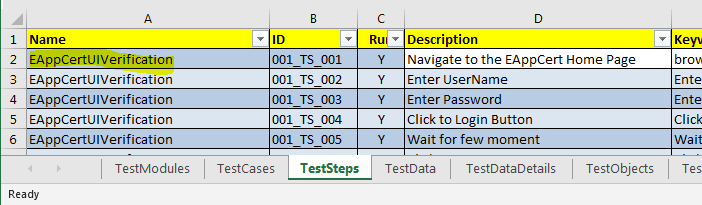
1. **‘ProjectName’**, **‘UserRequested’**, **‘Environment’** and **‘Release’:** User can enter value under **‘Value’** column corresponding to each attribute

2. **‘ChromeWaitPercentage’** and **‘FirefoxWaitPercentage**’: Value corresponding to these attributes must be in double digits and these values will be in percentage (%), for Chrome and Firefox browsers test execution these % value will be taken from the wait value in milliseconds provided in test case data sheet corresponding to ‘WAIT’ method.

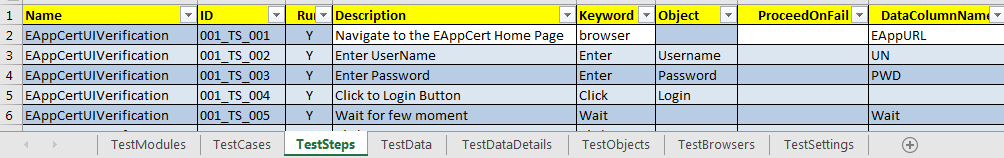
**d. TestCases:**

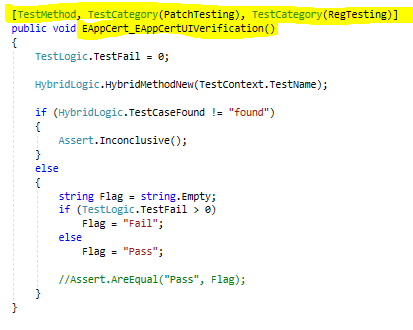


1. The ‘**TestCases’** sheet contains three columns ‘**Name’**, ‘**Description’** and **‘Run’**
2. The ‘**Name’** column contains Test Case’s for the application under test. Each Test Case test steps will be kept in ‘**TestSteps**’ sheet with its **Name**
3. User can add any number of Test Cases, by entering test cases in the ‘**Name’** Column which corresponds to ‘**TestMethod**‘name in the code

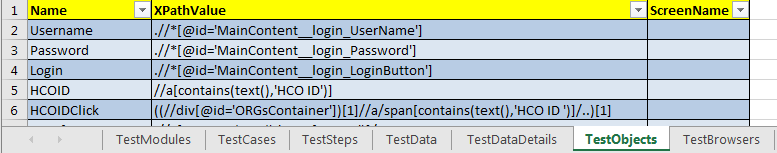


**e. TestSteps**

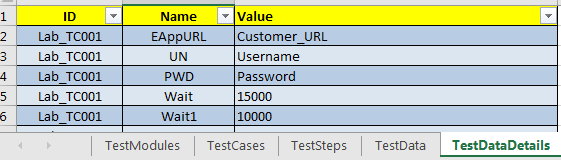




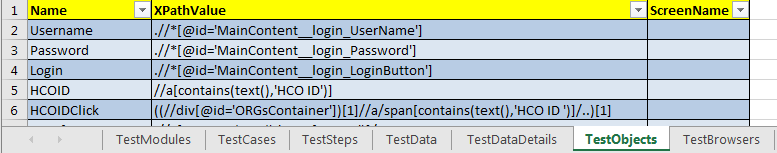
1. The **‘TestSteps’** name sheet contains columns **‘Name’,** **‘ID’**, **‘Run’**, **‘Description’**, **‘keyword’**, **‘Object’**, **‘ProceedOnFail’** and **‘DataColumnName’**
2. **‘ID’** column contains Test Step ID’s
3. **‘Run’** Column allow the User to enable/ disable the Test step by setting value to **‘N’** or **‘Y’**.
4. **‘Description’** column provide description of that test step
5. **‘Keyword’** column contains the action need to be perform over the Element/object, keyword here points to corresponding name in case statement in **HybridLogic.cs** file, which in turn corresponds to method in **SeleniumActions.cs**
6. **‘Object’** column contains the **‘ObjectName’** which have a describing name of the xpath value and it is corresponding to **‘XpathValue’** in ‘TestObject’ Sheet and **‘Name’** should be unique



1. **‘ProceedOnFail’** column allow the user to set the value **‘N’**. Where **‘N’** corresponds No, which denotes that if the test step fails, the script will come out of the test execution of that particular Test case at that test step and if user set its to **‘Y‘** for any particular test step and the step fails, the script continues with the execution next test step of that particular test case and the move to next test case in the queue.
2. **‘DataColumnName’** contains the data name, data name is used to enter the data in the given application which corresponds the **‘TestDataDetails’** sheet, **‘Name’** column which refer to **‘Value’** Column.

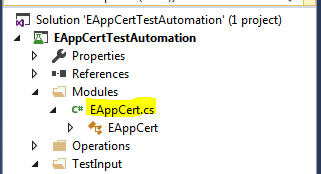


**f. TestObject:**

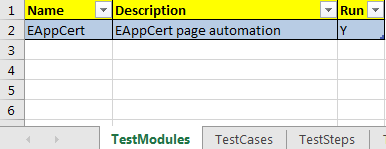


1. **‘TestObject’** sheet will contain three columns, **‘Name’**, **XpathValue, ‘ScreenName’**
2. **‘Name’** will have a describing name for the xpath value it is correspondingand **ObjectName** should be **unique.**
3. **XpathValue** will contain Xpath for the Element/ object for which the action as to be performed.

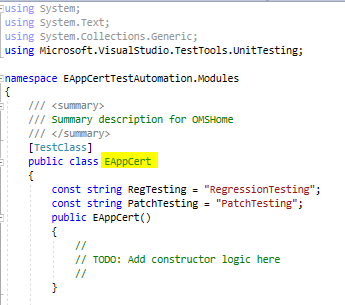
# **Modules**



1. The Modules Folder Consist of .cs files, these files are nothing but the Test Modules or a set of Test Modules of the application and we can include ‘n’ numbers .cs files within the folder Modules.
2. The Modules name must be the same name which was given in Test-Setting.xlsx file



1. Below is the code, the Test Class Name must be same name as the Module Name



1. User can add Test Case method within the Test class, the method should follow the naming convention as [**ModuleName\_TestCaseName**]



# **App.Config**

App.config is the configuration file used to configure below Parameters.



1. appName: refers to the module name

2. Password: refers to password

3. username: refers to user name

2. Customer\_url: refers to the internal URL

3. Employee\_url: refers to the external URL

4. Xml\_Location: is the place where XML file present

5. Report\_Location: is the location of the report folder

6. Masterreport\_location: location of the patch test report

# **Selenium Action (**[**SeleniumActions.cs**](#_Overview)**)**

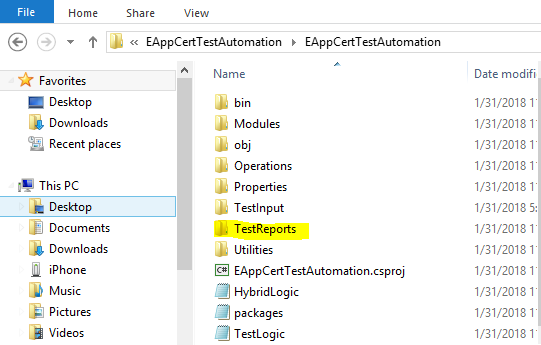
SeleniumActions is the class which contains all the methods, for the action to be performed on the object/ element on a given application

# **Hybrid Logic (**[**HybridLogic.cs**](#_Overview)**)**

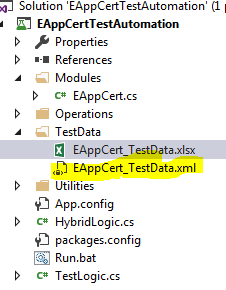
It’s the combination of data driven functionality and selenium actions.

# **Reports**

Details report are generated and saved in ‘TestReports’ folder under application test automation framework. Example below

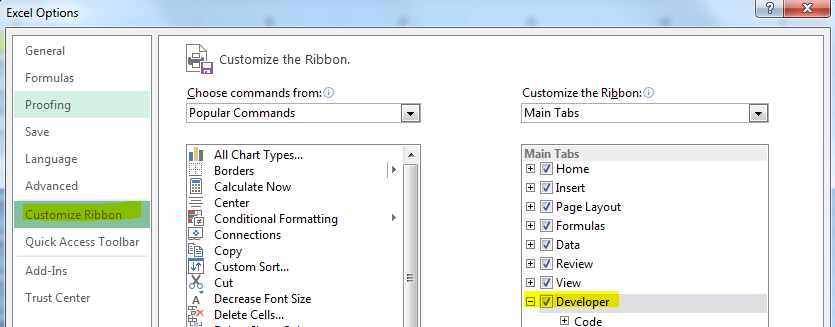


# **XML File**

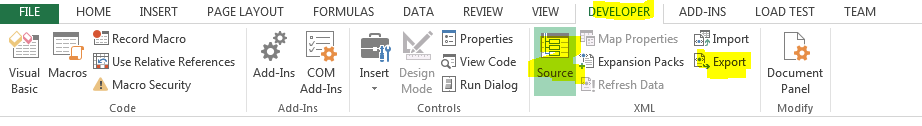


Steps to convert Excel file into XML

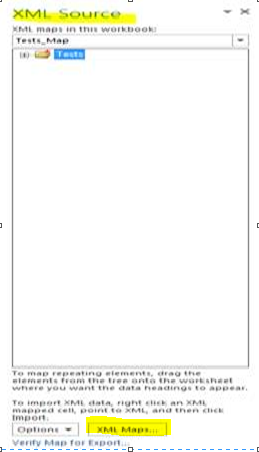
1. Open Excel file, click File  Options Customize Ribbon  Developer



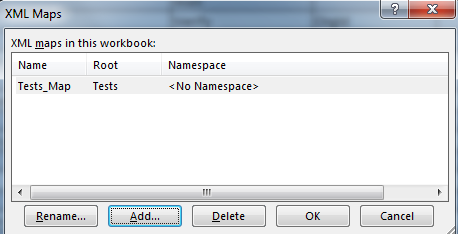
1. You should see ‘Developer’ tab now



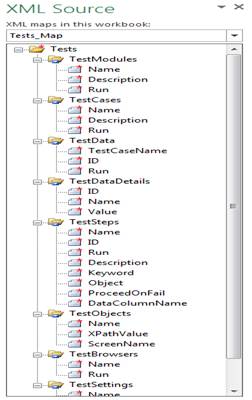
1. Click ‘Source’. You should see a new window on the right-hand side called ‘XML Source’



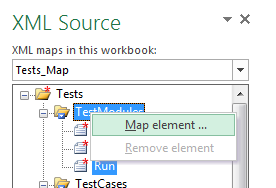
1. Click ‘XML Maps’ and click ‘Add’ to add the .xsd file to it



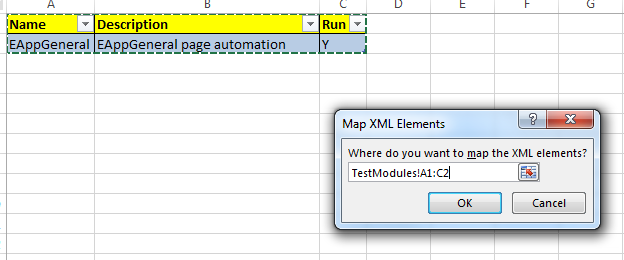
1. Click ok. It should load the XSD file in the ‘XML Maps’ window shown below



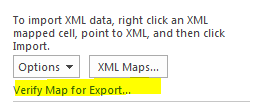
1. Right click each element and click ‘Map element …’



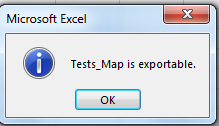
1. Now go the tab related to that element and select all the elements including the header and click ‘OK’



1. Repeat the same for all the elements, after done with all the elements, click ‘Verify Map for Export…’. This function makes sure the excel is mapped properly with xsd and is ready for export



1. If it is proper, you will get this message. Click ‘ok’. If not, you will need to check your mapping. Make sure right tab is mapped to right element



1. Now in Developer tab click ‘Export’ to export this into an xml file. You can point it to your existing xml file in the project. Make sure the file is checked out before you do this
2. Save the Excel file and close it
3. Check in the files into the project