# **Automated Unit Test** **Framework**

Contents

[**1.** **Introduction** 3](#_Toc428796455)

[**2.** **Setup AUT** 3](#_Toc428796456)

[**3.** **AUT in action – Execute a scenario** 12](#_Toc428796457)

[**4.** **AUT folder structure** 14](#_Toc428796458)

[**5.** **Developing Unit Test Case / Test Case / Scenario** 16](#_Toc428796459)

# **Introduction**

Automated Unit Test framework (AUT) is a framework tool used for automated unit testing, automated regression testing and end to end integration testing of IEDSS system. AUT is based on Selenium 2 Web driver API and is developed in Java. AUT consists of

* Test Suite of scenarios and test cases for testing IEDSS system and
* a framework to develop new unit test cases/test cases or update existing test cases

# **Setup AUT**

Required software

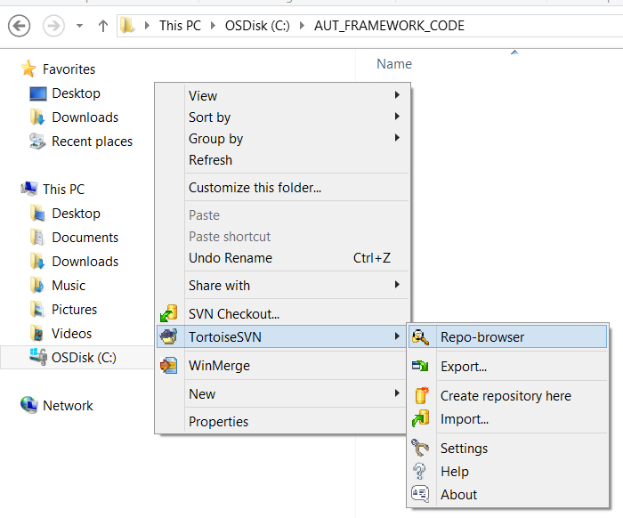
Please ensure that the following software is installed on your system before you proceed with setup.

1. Java JDK 7.x or JRE 7.x or higher version
2. Eclipse (Luna or higher versions) or MyEclipse Blue Edition
3. TortoiseSVN version 1.8.7 or higher - [download](http://tortoisesvn.net/)
4. Internet Explorer or Firefox

Setup

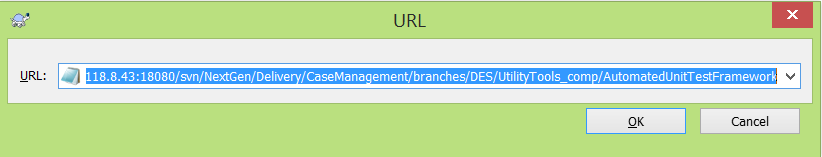
Please follow the following steps to setup AUT on your system.

**Step 1:** Create a folder in C drive (C:/AUT\_FRAMEWORK\_CODE). Right click inside the folder and Select TortoiseSVN and from there Repo-browser as shown below.

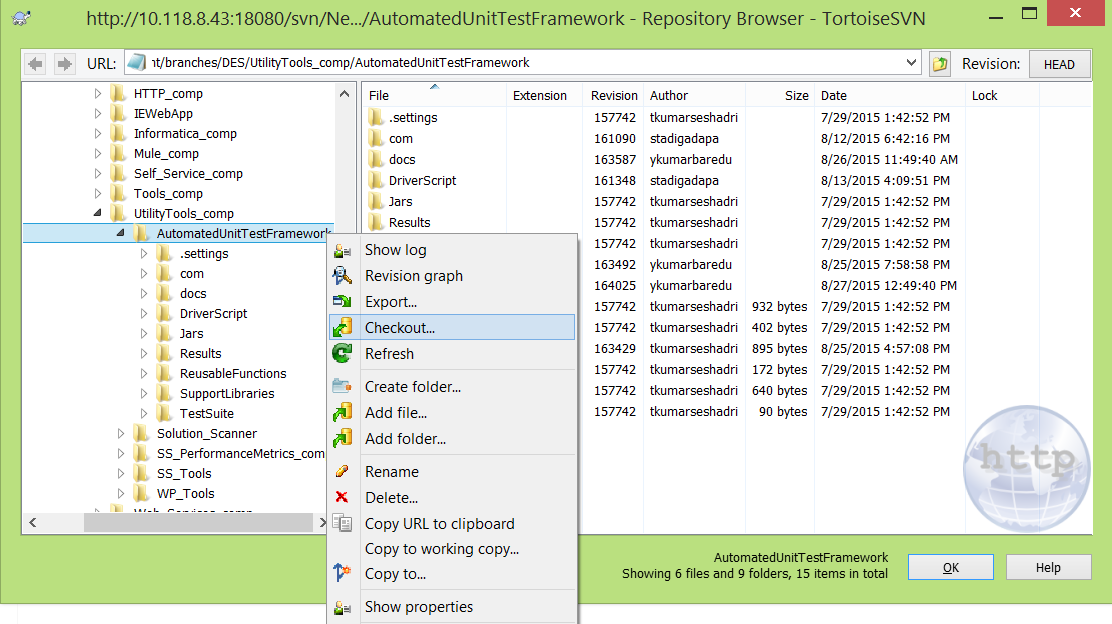


**Step 2:** Give the URL of the SVN code and click on Ok button.

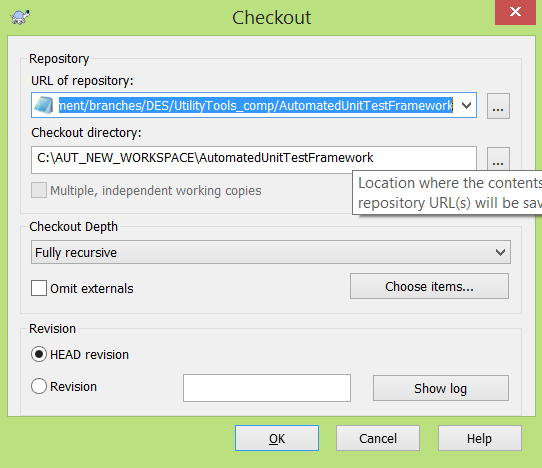
**SVN URL:** <http://10.118.8.43:18080/svn/NextGen/Delivery/CaseManagement/branches/DES/UtilityTools_comp/AutomatedUnitTestFramework>



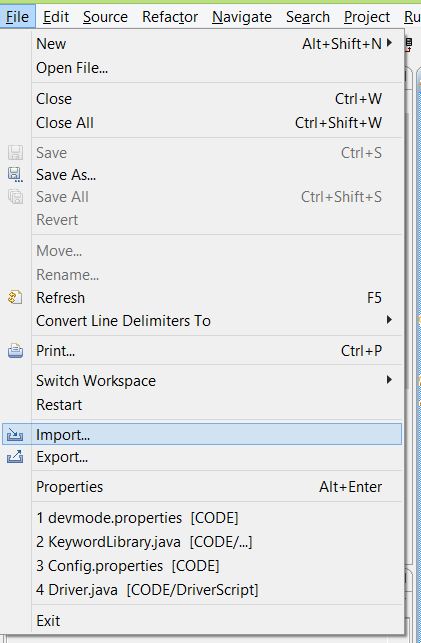
**Step 3:** After clicking on Ok button, pop up will come with the folders. Right click on the AutomatedUnitTestFramework folder as shown in the screenshot and click on checkout option.



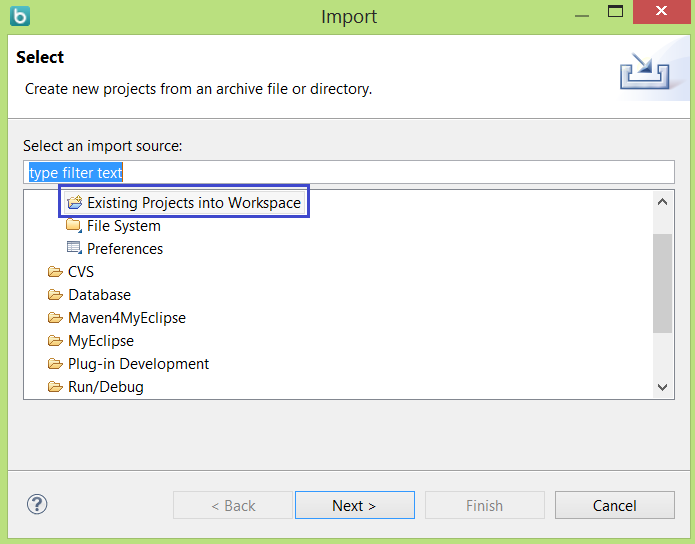
**Step 4:** On clicking the Checkout option, one popup will appear with the repository URL and the Checkout directory (this should be the current folder).



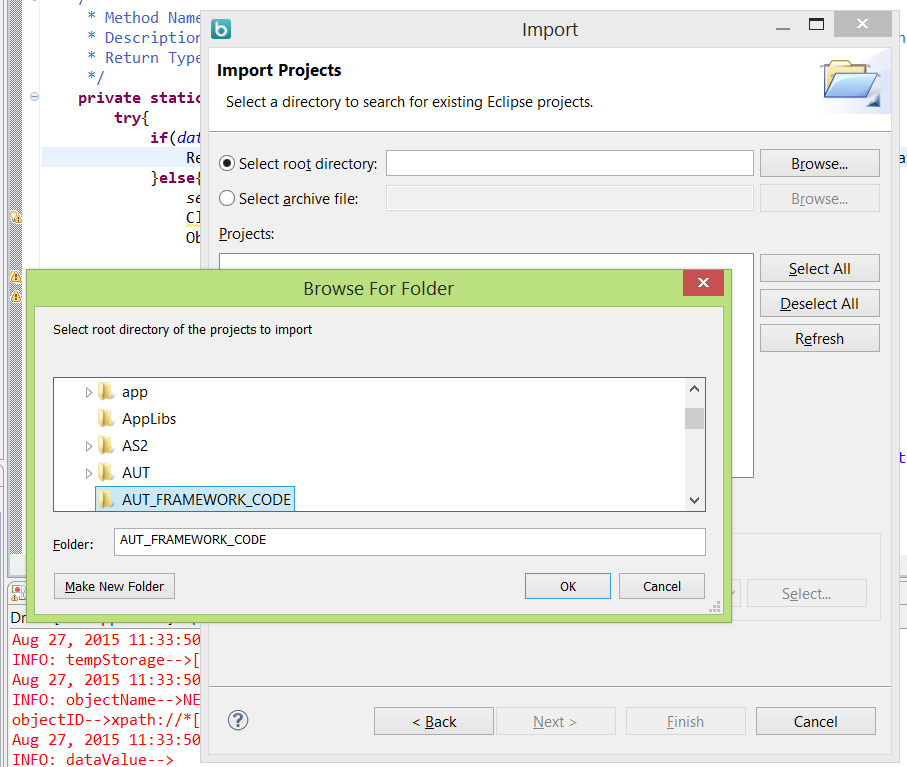
**Step 5:** Open MyEclipse and navigate to the Import option (File -> Import).



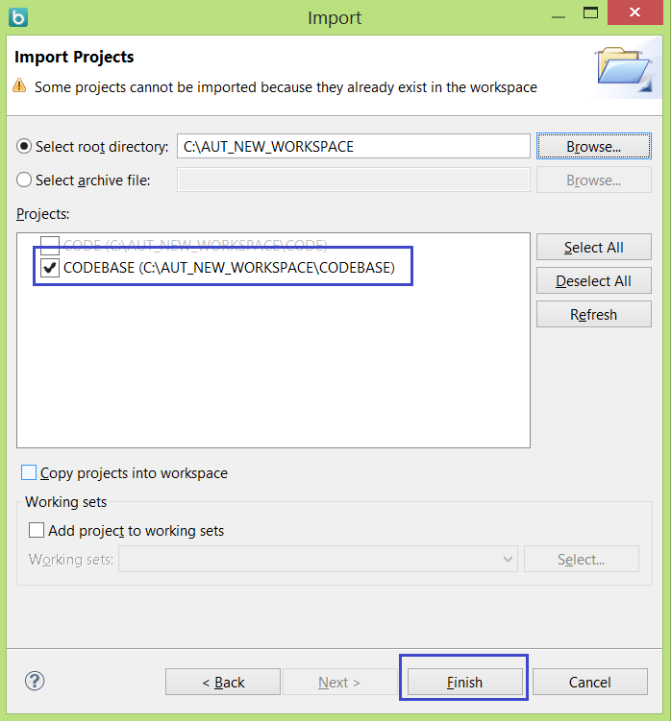
**Step 6:** Select **Existing Projects into Workspace** sub option from **General** category and click on Next.



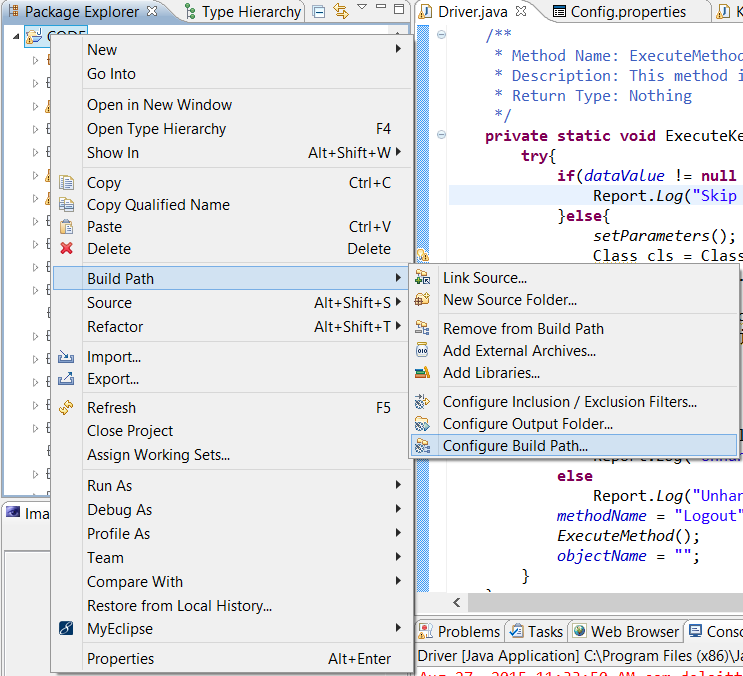
**Step 7:** Import Projects window will appear. Click on Browse button and select the location of the folder where the project is checked out. Click OK after selecting the folder location.



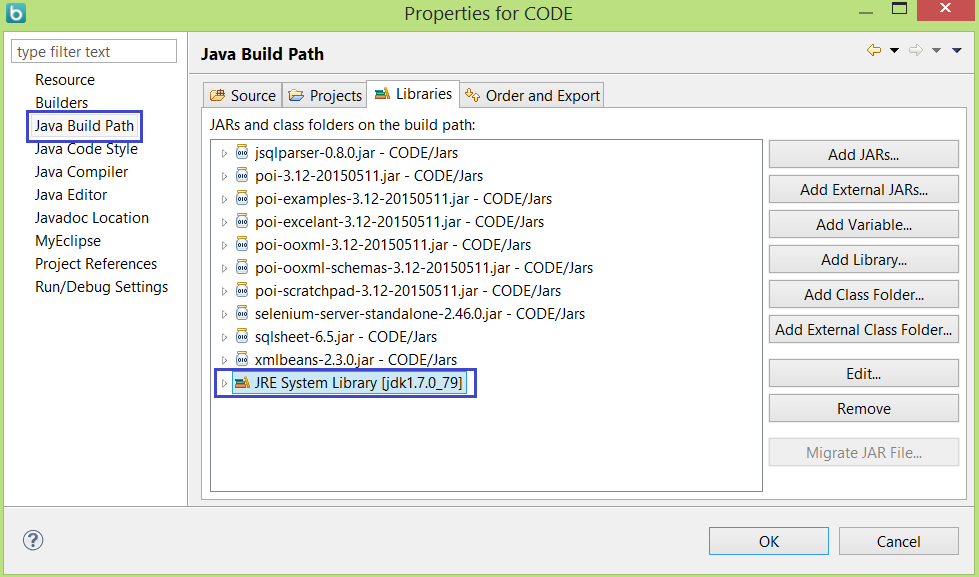
**Step 8:** Once the Ok button is clicked the checked out code will show up under the Projects location. Select the particular location and click on Finish button.



**Step 9:** After adding the project in workspace, build path needs to be configured. Right click on the project (which is present in package explorer), in the Build Path option click on **Configure Build Path**.

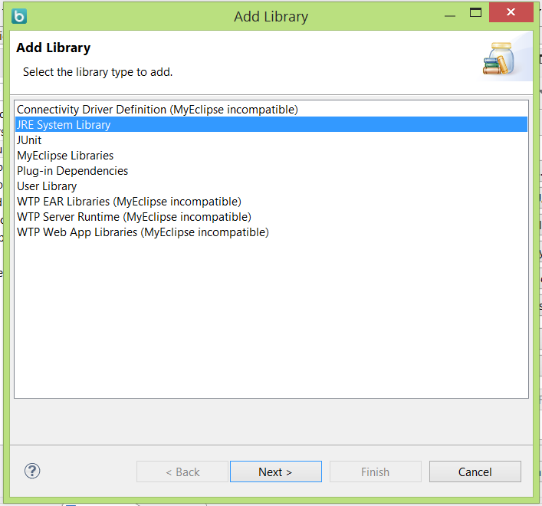


**Step 10:** Change the build path to the JDK (**JDK version should be 1.7**) location.

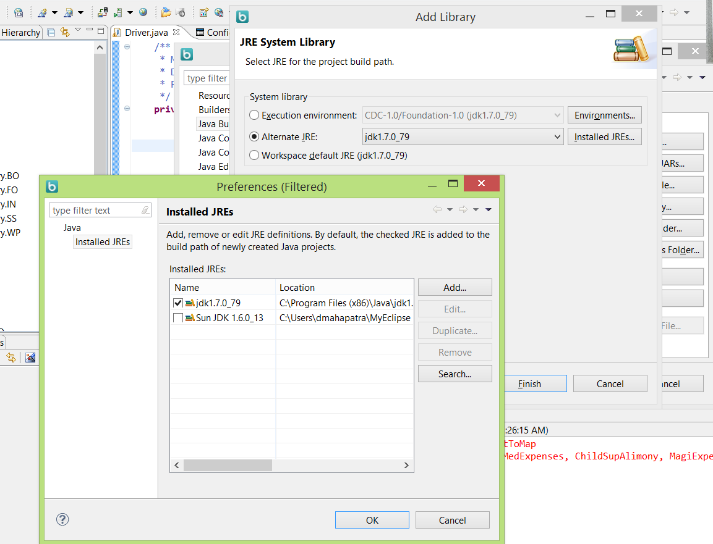


If the Java7 library is not present, follow following steps to set up system library.

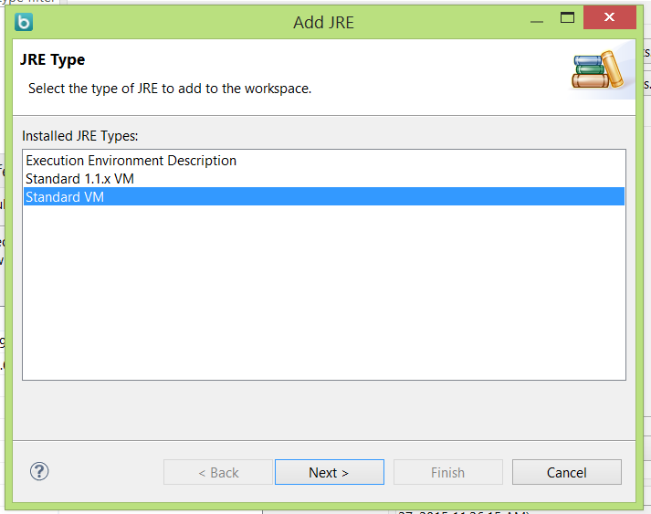
* Click on Add Library option in the same popup. Select **JRE System Library** and click on **Next** button.



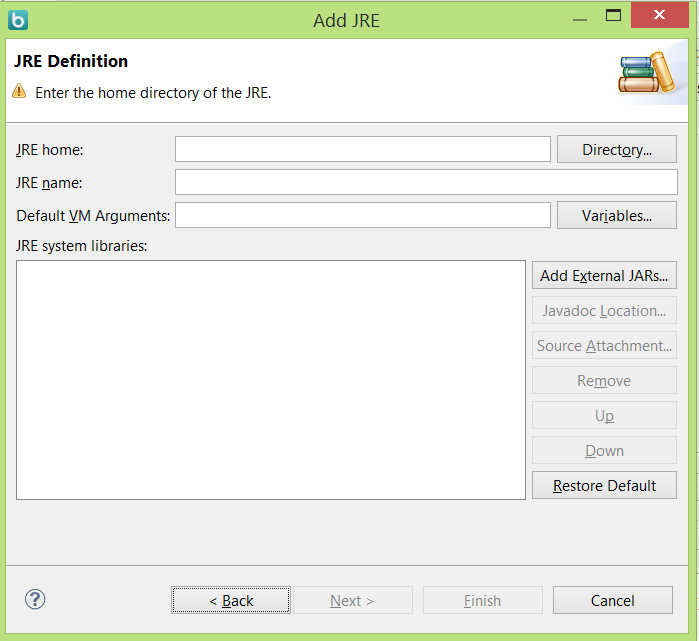
* Another pop up will open for Adding the library. Select **Alternate JRE** and click on **Installed JREs** button. One popup for **Preferences** will open. Click on the **Add** button.



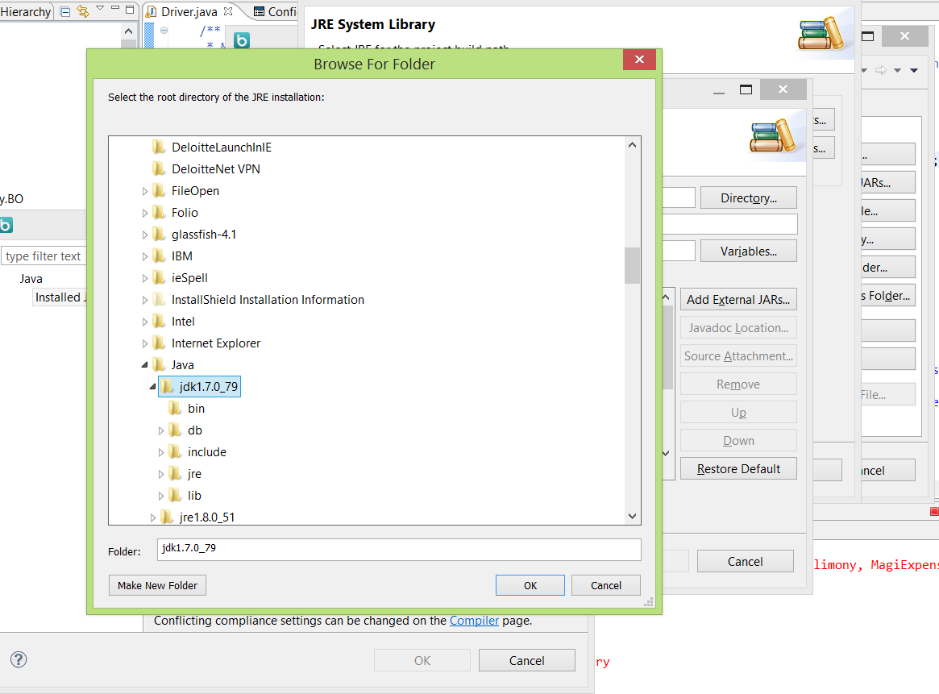
Popup to **Add JRE** will be opened.



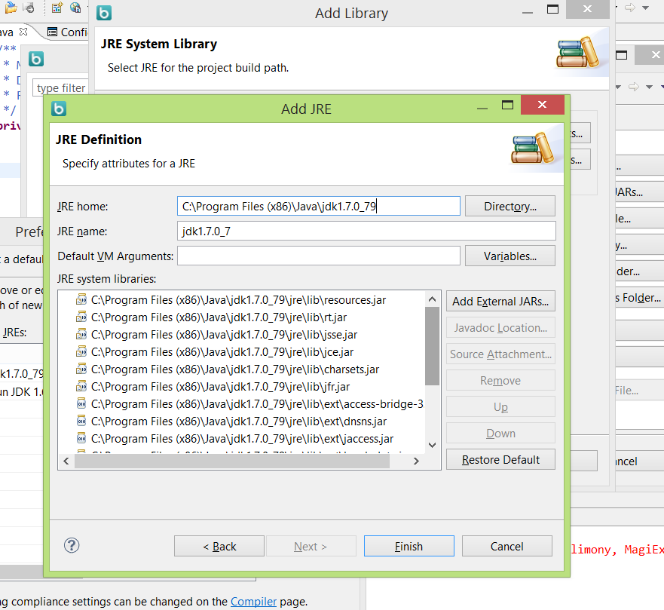
* Click on **Standard VM** and click on **Next** button. On click one popup will open to define the installed JRE location.



* Click on **Directory** and select the folder location where Java 7 is installed. Click **OK** after that.

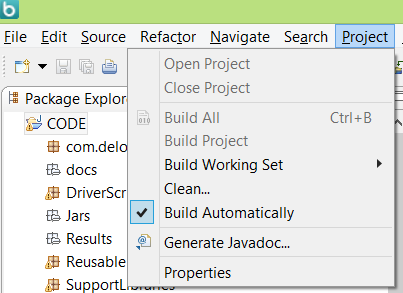


* Existing Jars of Java 7 should appear in the **Add JRE** popup. Now click on **Finish** button.

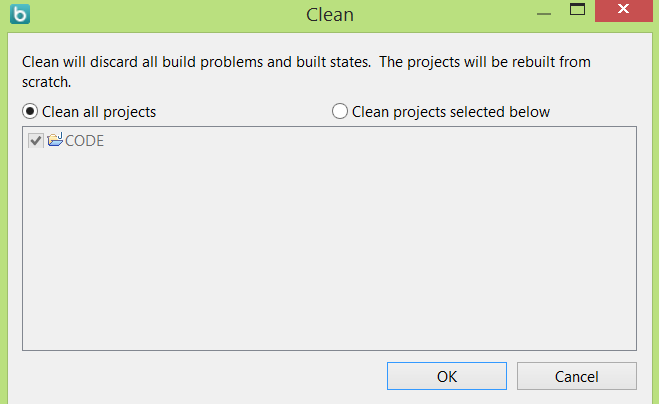


Click on Ok and refresh the project (click on the added project and press F5 or right click on the project and select **Refresh** option).

**Step 11:** Click on the project option in MyEclipse and check Build Automatically option.

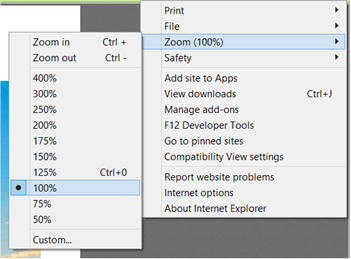


**Step 12:** Click on **Clean** option above Build Automatically. Select Clean all projects or Clean Projects selected below (and select the AUT project) and click on OK button.

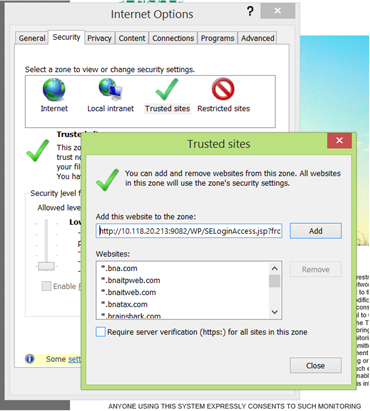
****

**Step 13:** Change the Internet Explorer setting as follows –

* Set the zoom percentage to 100%.

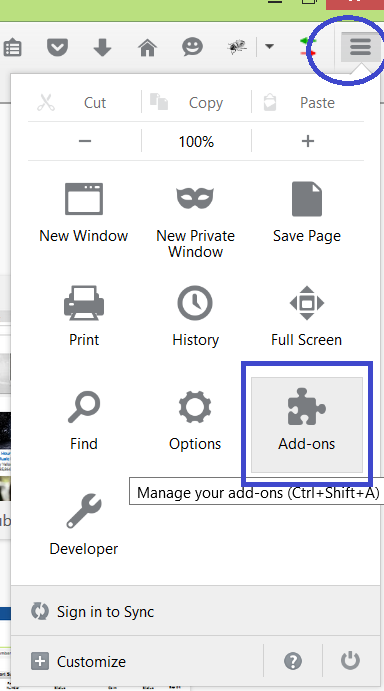


* Go to Tools -> Internet options -> Security tab -> click on Trusted Sites -> Click on Sites button -> paste the Site link and click on Add button -> close the popup -> Click on OK button of Internet Options popup.



**Step 14: Install Firepath add on.**

* Click on the menu button to open the menus from Firefox and click on Add-ons button.



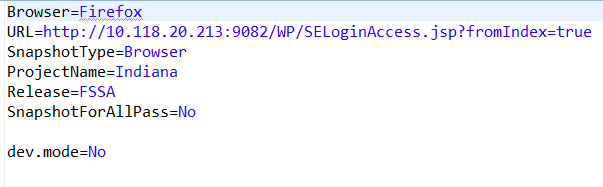
* On the Add-ons page search for Firepath. And click on the Install button and restart the browser after installing it.



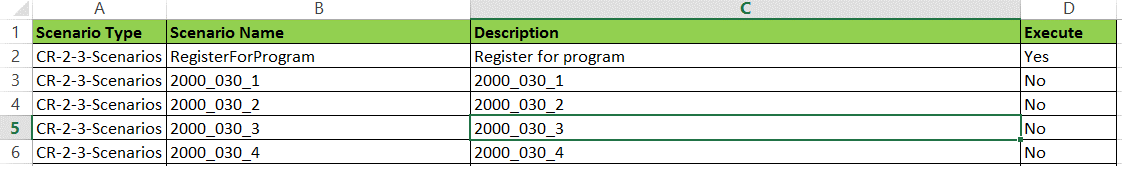
# **AUT in action – Execute a scenario**

Please follow the below mentioned steps to execute a scenario on your system.

**Step 1:** Open **config.properties** file and change the Browser (‘IE’ for internet explorer and for Firefox write ‘Firefox’) and URL as below (here the SNET URL is provided, best practice is to take latest code and set up the code in local and provide the local URL of application)-

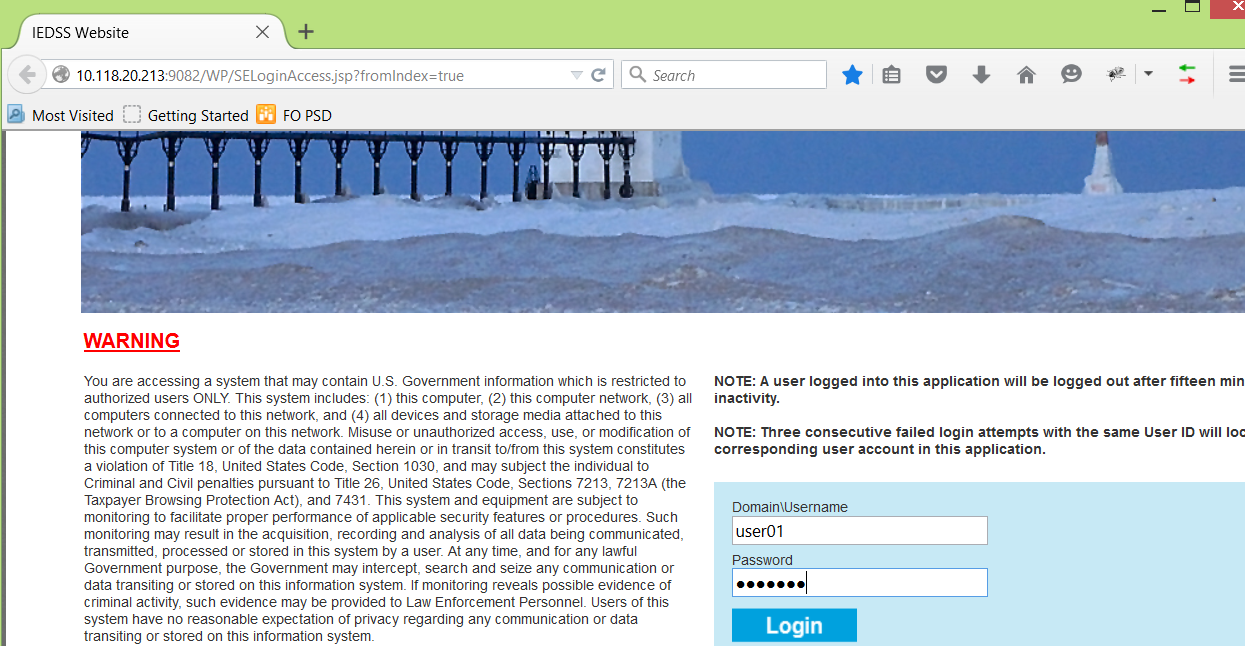


**Step 2:** Open **RegressionScenarios.xlsx** file (in TestSuite Folder) and for **Register for program** scenario mark the **Execute** column as **Yes**. Mark other scenarios’ **Execute** column as **No**.



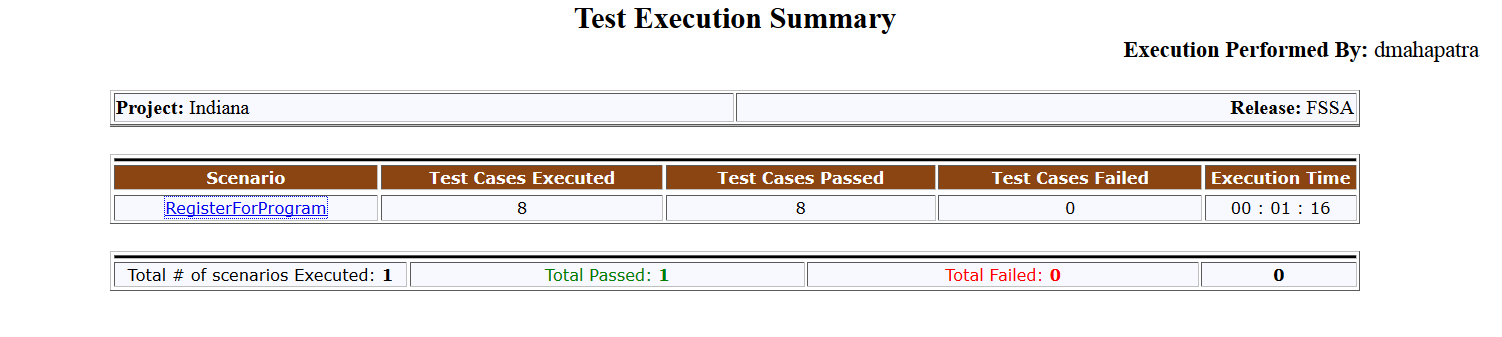
**Step 3:** Run the application by double clicking on StartExecution.bat file (or by running Driver.java file as Java application).

**Step 4:** Depending on the browser set in config.properties file, Internet Explorer or Firefox will open with the provided URL. And it will login to the application.



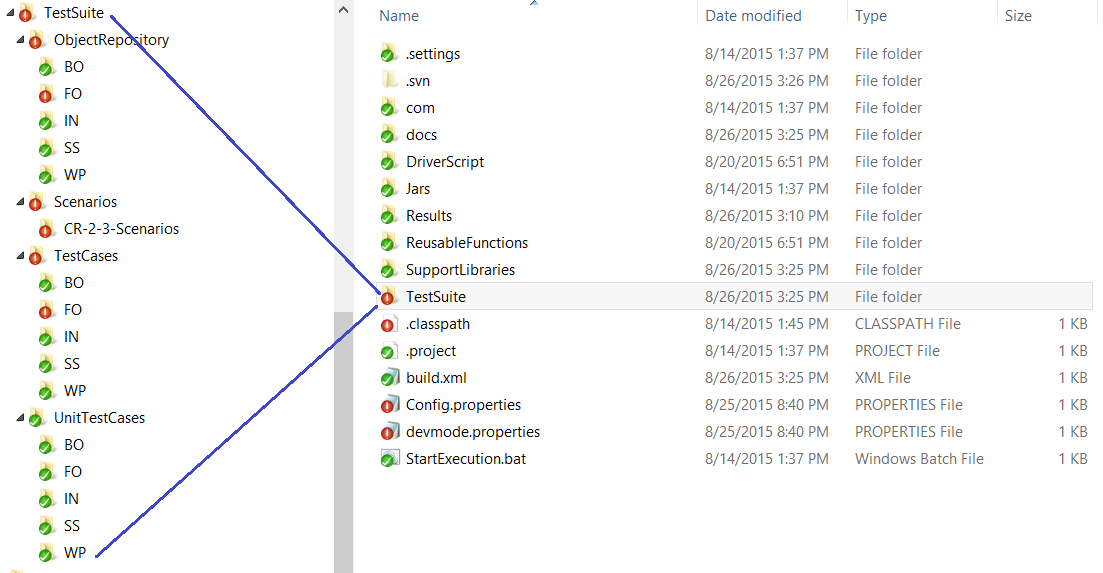
**Step 5:** Check the Results folder to check the status of the Run.

Inside Results folder 2 folders (**Run** and **RegressionRun**) will be generated after the execution gets completed. Run folder contains the detailed results for each test scripts and RegressionRun folder contains the summary of the scenarios. Open **RegressionSummary.htm** inside RegressionRun folder to check the status of each steps.



# **AUT folder structure**

Folder structure of AUT -

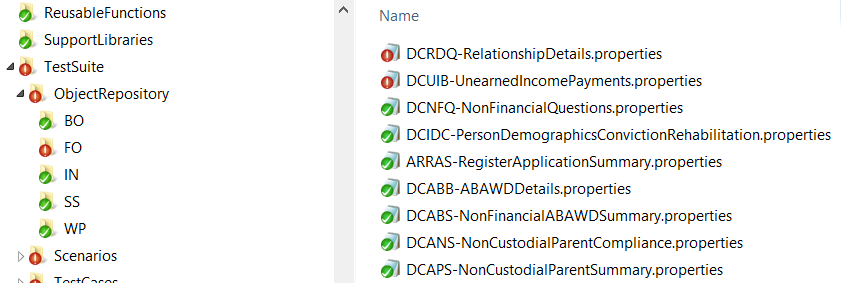


TestSuite is one of the main and important folder. This contains ObjectRepository, Scenarios, TestCases and UnitTestCases folder and RegressionScenarios script (this contains the executable script names).

As shown in the above screenshot - ObjectRepository, TestCases, UnitTestCases folders contain subfolders (BO, FO, IN, SS, WP) which are the separate modules of the overall project. The subfolders are created so that the respective modules are placed in the subfolders.

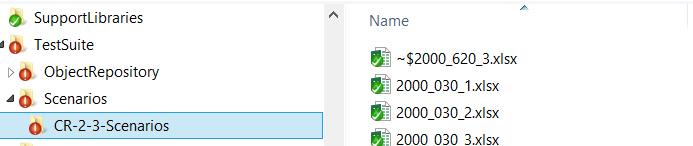
ObjectRepository:

This folder contains the properties files of every modules. Which contains the variable name and the xpath of the page web elemnts.



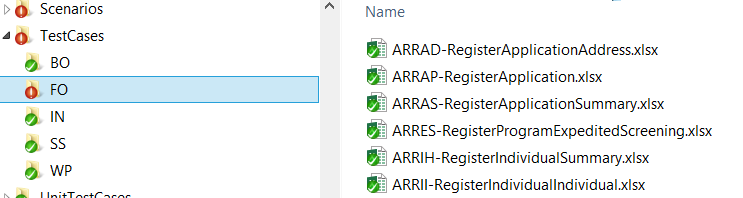
Scenarios:

This folder contains the different scenario scripts for the different release of CR (as an example – CR-2-3-Scenarios). The scenario scripts contain the testscript name and the data. PFB the screenshot -



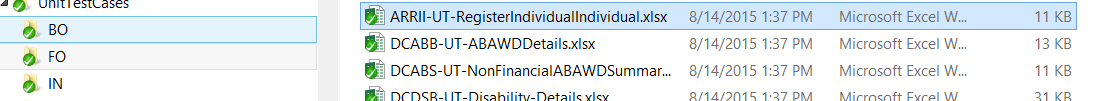
TestCases:

This contains the test scripts for each module.



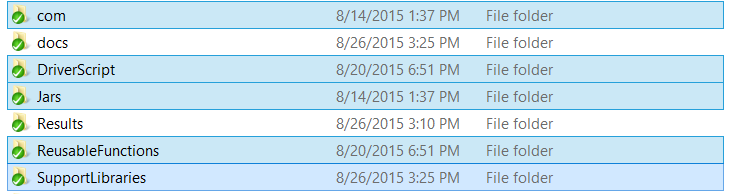
UnitTestCases:

This folder contains the unit test scripts for the different modules. Scripts are same as test cases scenarios but the scenarios and flows will be different depending the unit test case.



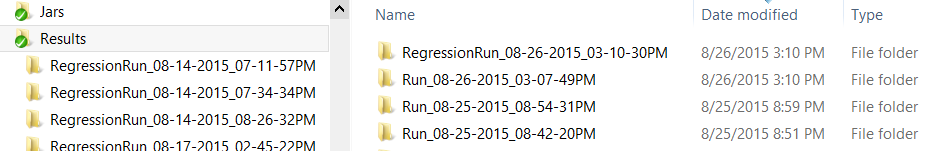
Java source files and folders:

Following highlighted folders contain java classes.



Results:

Results folder contains the results of each run. For each run **RegressionRun** folder and **Run** folder will be created with the timestamp as a suffix to these folders. RegressionRun folder contains RegressionSummary.htm file which contains the scenario name, executed test cases, passed and failed test cases count and total execution time for the script.



Important files:

* **Config.properties –** This file is for configuring the properties of the application.
* **devmode.properties -** This file will be useful when the value of dev.mode property will be ‘Yes’.
* **StartExecution.bat –** This file is present to run the application.

# **Developing Unit Test Case / Test Case / Scenario**

Please find below the step by step process on how to create a Unit Test Case and test the result of the script.

Here we are going to create unit Test Case for **Register Individual** screen.

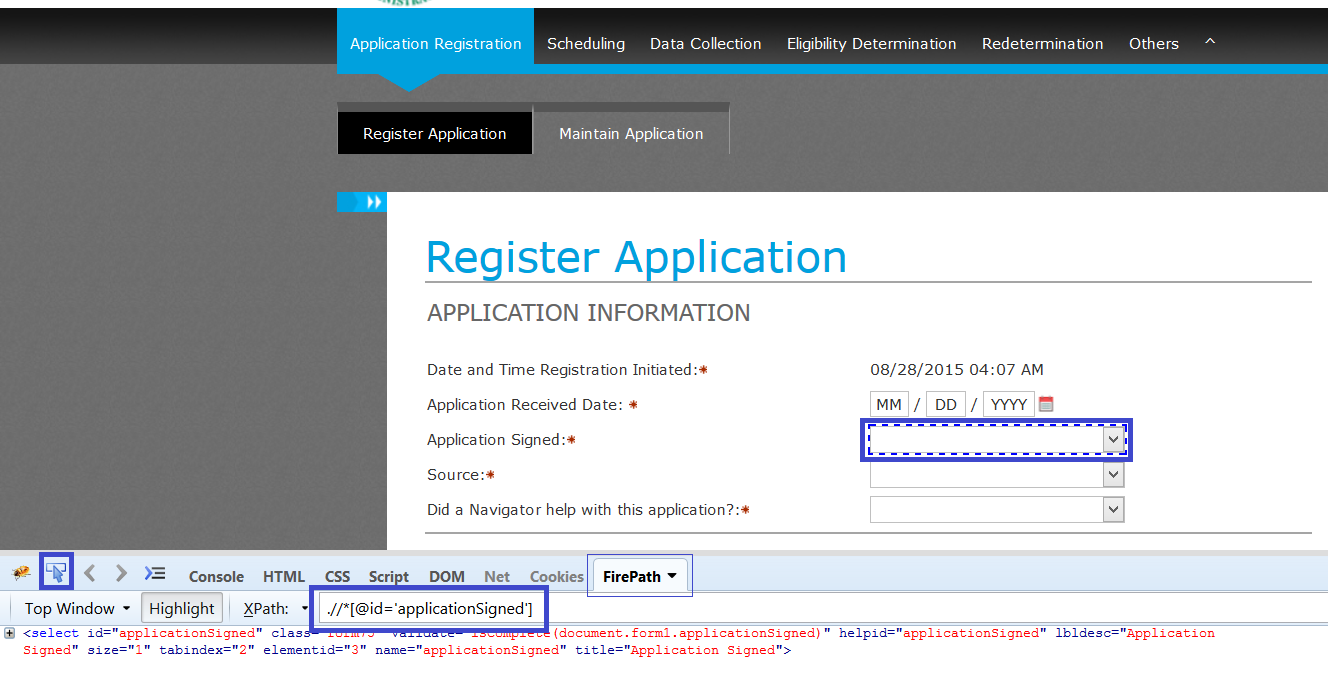
Developing Unit Test Case

**Step 1:** **Create Object Repository –**

All the web elements of a particular page is mapped with the xpath of the properties file. Create a .properties file in the desired module of ObjectRepository folder. Naming convention of the properties file should be like –

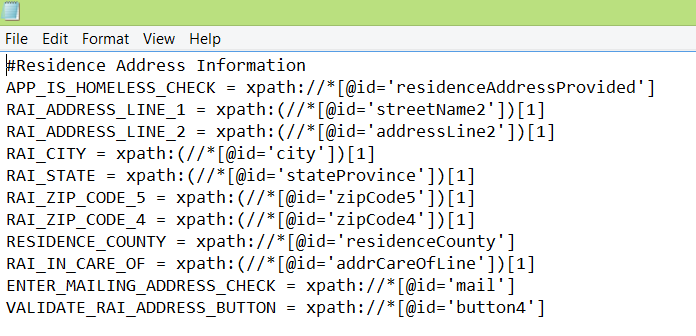
<PAGE\_ID>-<Page Name>.properties (example- **ARRAD-RegisterApplicationAddress.properties**)

Find out the xpath of the element using firepath.



First click on **inspect an element** and check the **FirePath** of the particular element as shown in the above screen shot.

Give a unique variable name for a particular element and give the xpath of the element as below.



Create the properties file for all the elements.

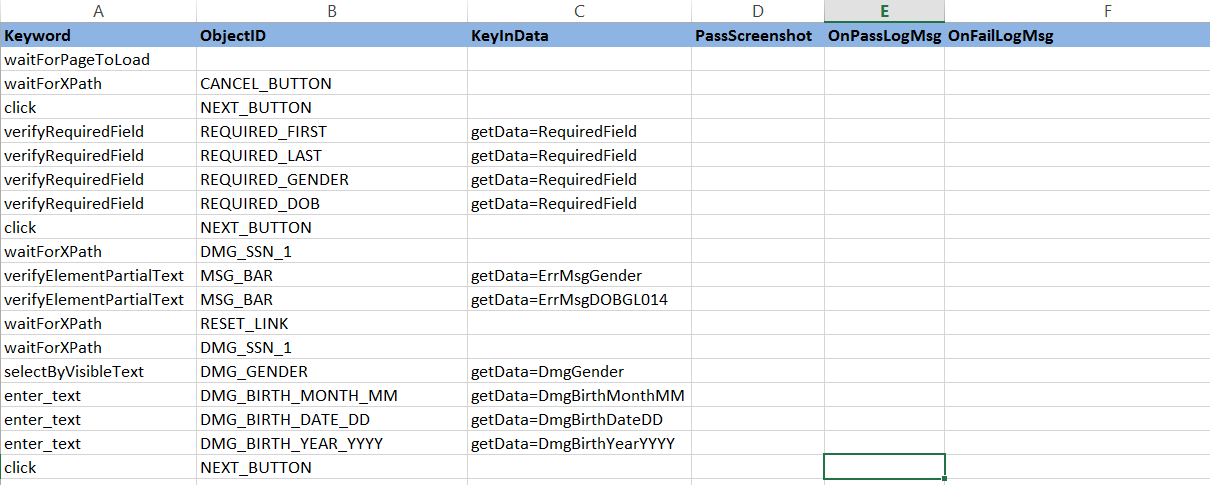
**Step 2:** **Create Unit Test Cases or Test Cases–**

Create a .xlsx script for Unit Test Cases with name as - <PAGE\_ID>-UT-<Page Name>.xlsx (example- **ARRII-UT-RegisterIndividualIndividual.xlsx**)

Create a .xlsx script for Test Cases with name as - <PAGE\_ID>-<Page Name>.xlsx (example- **ARRII-RegisterIndividualIndividual.xlsx**)

Script contains two tabs.

* **TestSteps tab** – This contains the colums as-> keyword (which is function in the KeywordLibrary.java), ObjectID (This is the variable name provided in the properties file) for a particular element, KeyInData (This contains the getData=<data value> from the TestData tab in the same excel), PassScreenshot (This contains the value either Yes or No which will take the screenshots on the pass of a value) and OnPassLogMsg and OnFailLogMsg which will contain the message for pass and fail conditions.



You can refer to [KeyWord Library.xlsx](http://10.118.8.43:18080/svn/NextGen/Delivery/CaseManagement/branches/DES/UtilityTools_comp/AutomatedUnitTestFramework/docs/KeyWord%20Library.xlsx), for the complete list of key words available in AUT.

* **TestData tab** – This contains the iteration number which can be run by the application. And the values of the field based on the getData value of the particular field.

As an example the get data value of the field should be described in this field.

As per the screenshots in the TestSteps tab,

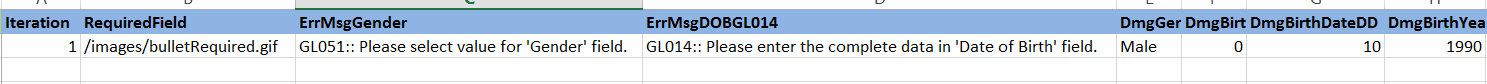
keyInData of CSCD\_DATE\_MM is getData=CSCDDATEMM

Create one Column named CSCDDATEMM in the TestData tab and enter the value.

Important points to remember –

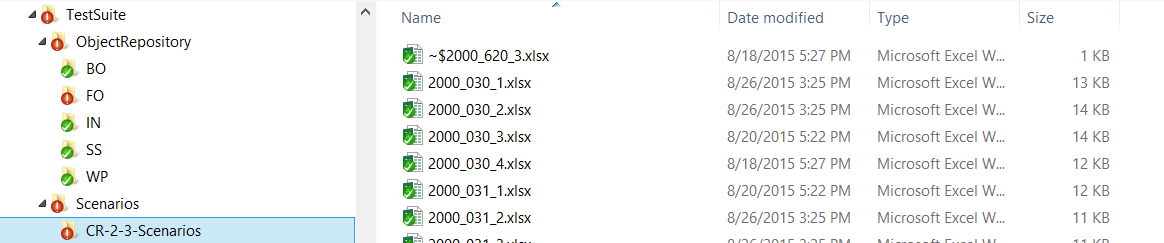
1. To skip any field enter the data as – n/a

2. Make sure there is no extra space in both the tabs. This manual error could result the script to stop its execution.



**Step 3:** **Compose scenario –**

Inside **Scenarios** Folder we can create scenario scripts (.xlsx file). You can name your scenario files appropriately and place it in this folder.



This script contains **TestCases** tab –

* Test Cases – This tab contains **BusinessModule, ObjectRepository, TestScript, Description, Execute, NumberOfIterations, StartIndexForIteration** columns.

**BusinessModule –** Contains the subfolder or module name where the script is present.

**ObjectRepository –** Contains the properties file name.

**TestScript –** Contains the Test Script or unit Test Script name.

**Description –** Description of the script.

**Execute –** Value can be ‘Yes’ or ‘No’. Depending on the value the script will be executed (‘Yes’ will be executed and ‘No’ will not be executed).

**NumberOfIterations –** The value of this column defines how many time the script will be executed.

**StartIndexForIteration –** Depending on this value the scenario will start getting executeed from that index number of the script.

As an example, if the value is 2, then the values from 2nd row of TestData tab of the particular script will be taken while execution.

**Step 4: Creating Regression Script –**

Inside TestSuite folder we need to update the Scenarios tab, which contains the following –

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario Type** | **Scenario Name** | **Description** | **Execute** |
| CR-2-3-Scenarios | RegisterForProgram | Register for program | Yes |
| CR-2-3-Scenarios | 2000\_030\_1 | 2000\_030\_1 | No |

**Scenario Type** – Contains the sub folder name inside Scenario folder.

**Scenario Name** – Contains the Scenario script name.

**Description** – Contains the description of the particular scenario run.

**Execute** – Depending on the value of the column (Yes or No) the script will be executed.