Contents

[1 Readiness steps for automation testing 1](#__RefHeading___Toc488_2818620495)

[1.1 Prerequisites 1](#__RefHeading___Toc490_2818620495)

[1.1.1 Installation Set 1 1](#__RefHeading___Toc492_2818620495)

[1.1.2 Installation Set 2 2](#__RefHeading___Toc494_2818620495)

[1.1.3 Installation Set 3 3](#__RefHeading___Toc496_2818620495)

[1.2 Preparing to run automation tests 8](#__RefHeading___Toc498_2818620495)

[1.2.1 Settings 8](#__RefHeading___Toc500_2818620495)

[1.2.2 Controlling run parameters 9](#__RefHeading___Toc502_2818620495)

[1.2.3 Running the Automation script 11](#__RefHeading___Toc504_2818620495)

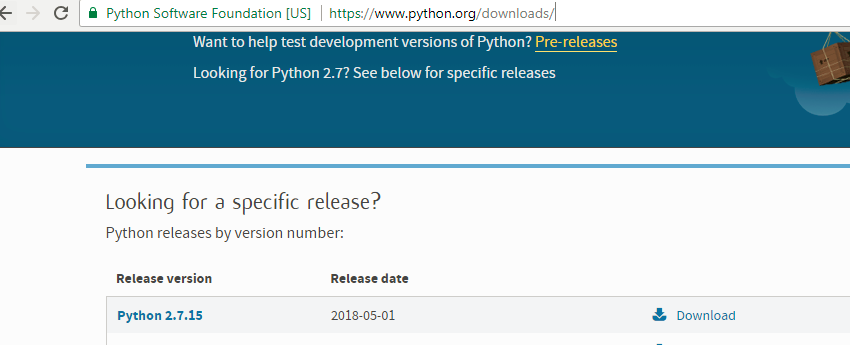
[1.2.4 Log file and Screenshots 11](#__RefHeading___Toc506_2818620495)

# Readiness steps for automation testing

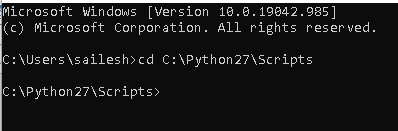
## Prerequisites

### Installation Set 1

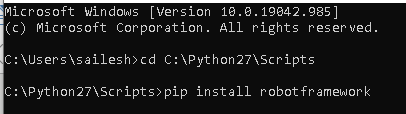
#### Navigate to <https://www.python.org/downloads/> and click on the download button to install Python 2.7



#### Open command prompt and change the directory to C:\Python27\Scripts by running the command ‘CD C:\Python27\Scripts’



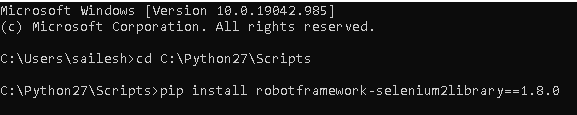
#### Install robot framework using: pip install robotframework



### Installation Set 2

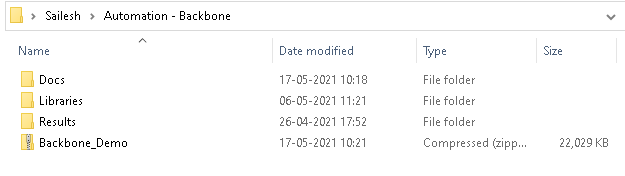
#### Install selenium2library using: pip install robotframework-selenium2library

1. pip uninstall robotframework-selenium2library
2. pip install robotframework-selenium2library==1.8.0



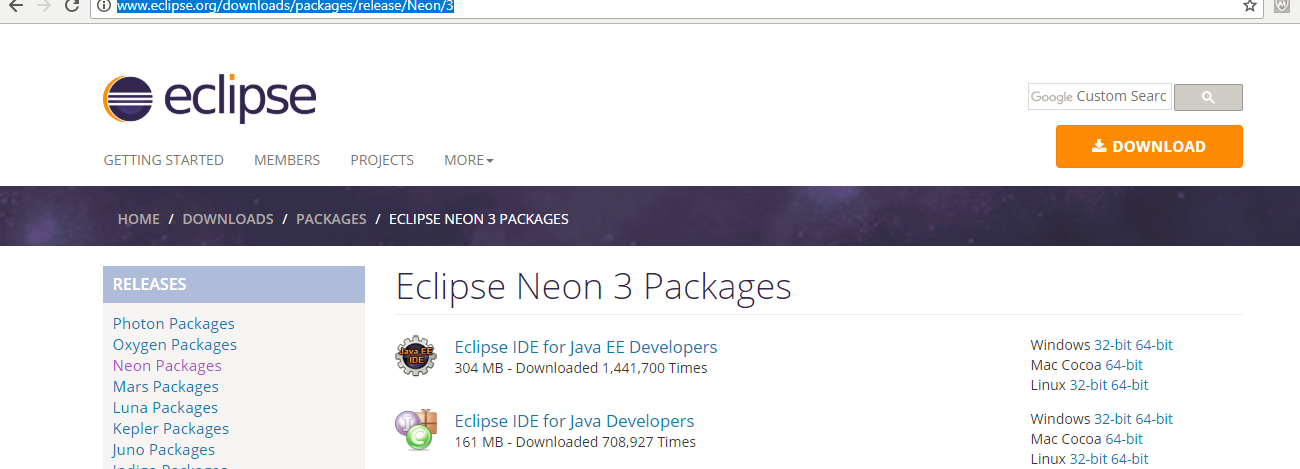
#### Copy the framework zip file to your local system

E.g. *C:\Users\sailesh\Desktop\Sailesh\Automation - Backbone*

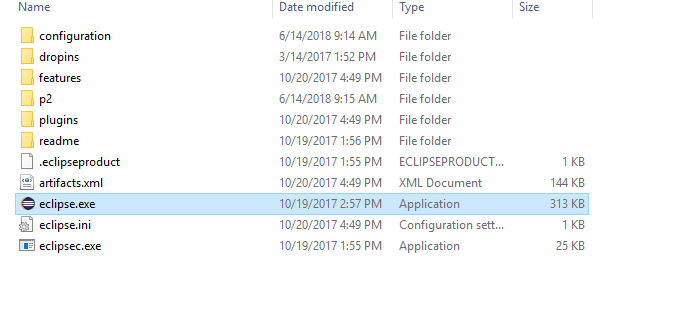


### Installation Set 3

#### Install Eclipse IDE for Java Developers by navigating to <http://www.eclipse.org/downloads/packages/release/Neon/3> and click on download to install it

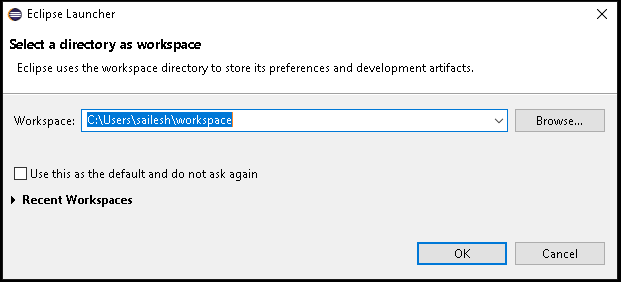


#### Open Eclipse by clicking on the ‘eclipse.exe’ file



#### Eclipse Editor will ask you to select the directory. Select the directory where the framework files are stored in the step 6 and click on OK

*E.g. C[:\Users\sailesh\](file:///D:/)*

**

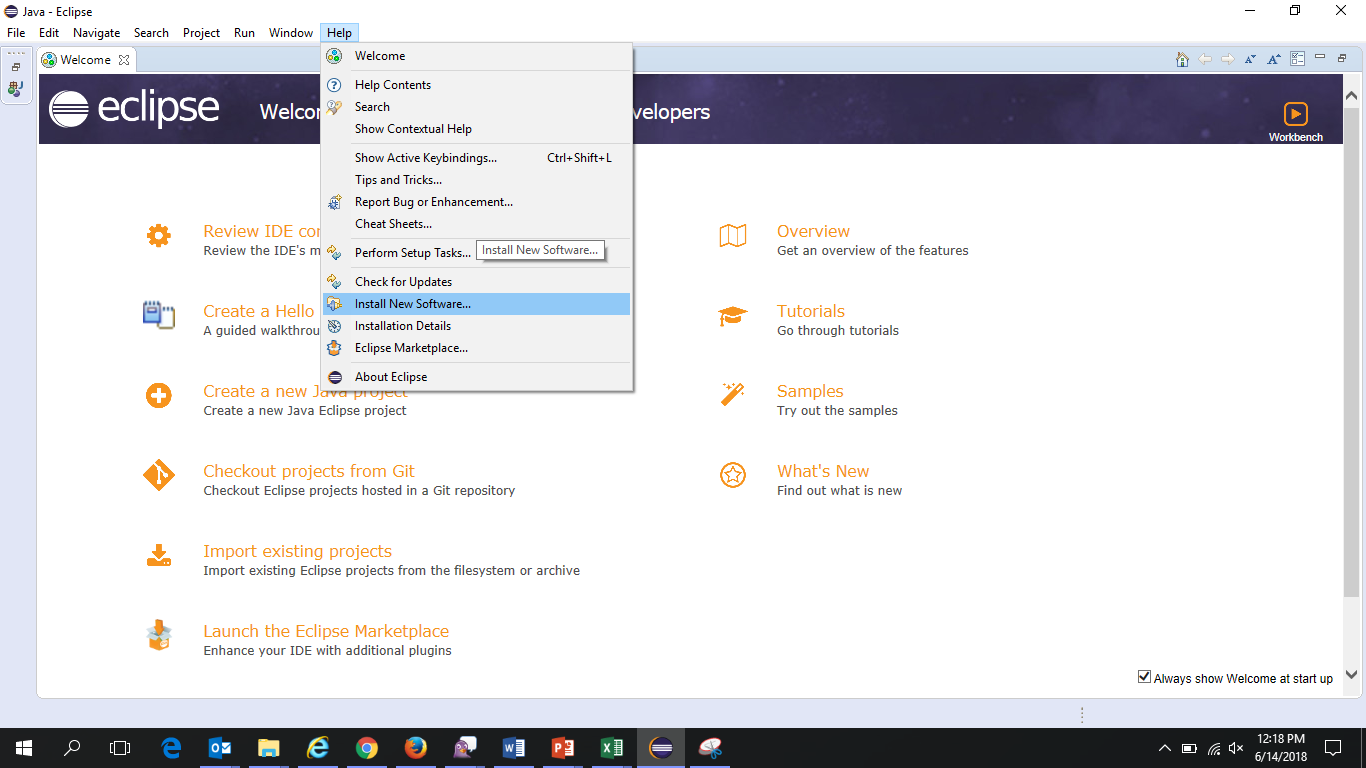
#### Eclipse will open and will display the home page as shown below



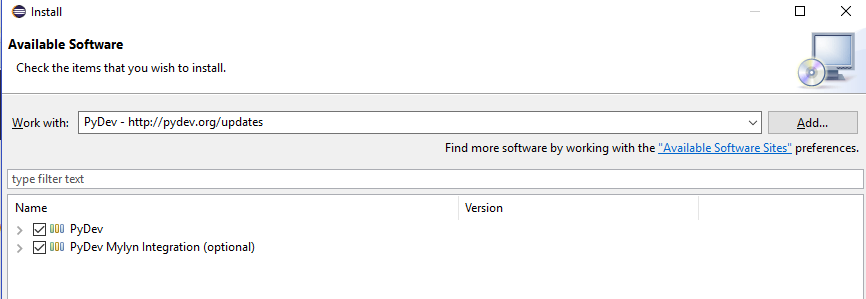
#### Install the following software using Help -> Install New Software…

*a. PyDev - https://www.pydev.org/update\_sites/8.0.1/*

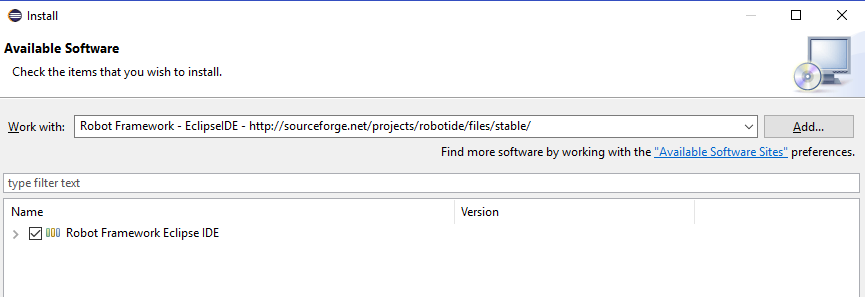
*b. Robot Framework - EclipseIDE - http://sourceforge.net/projects/robotide/files/stable/*



*a. PyDev - https://www.pydev.org/update\_sites/8.0.1/*

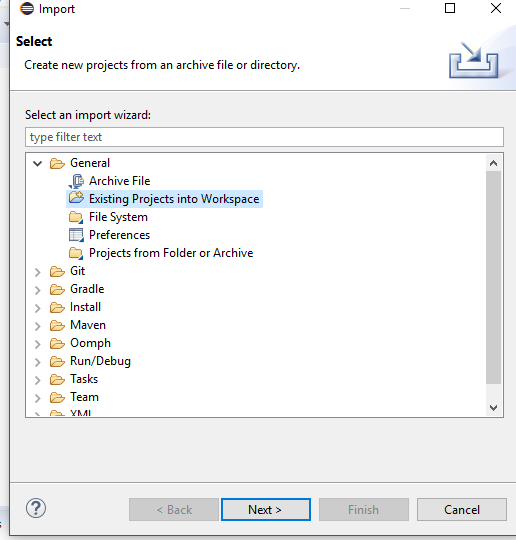


*b.Robot Framework - EclipseIDE - http://sourceforge.net/projects/robotide/files/stable/*

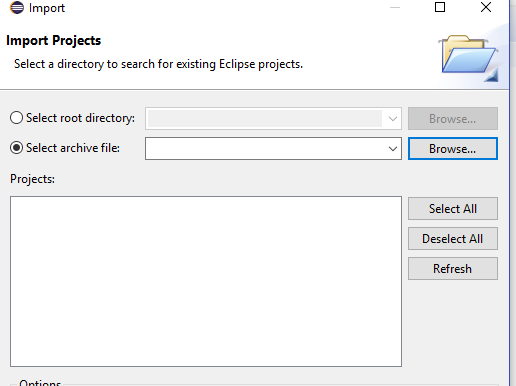


#### If the installation fails, open the robottide folder, close your Eclipse and copy the first jar -file to your Eclipse's features directory and the second jar -file to your Eclipse's plugins directory. You can find the feature & plugin directories under your Eclipse installation directory, eg. /eclipse/features and /eclipse/plugins.

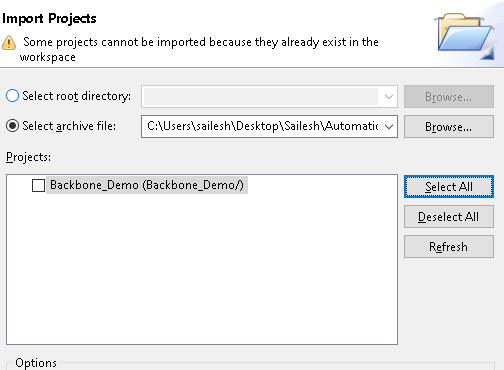
#### Import the zip file shared by automation team as Existing Projects into Workspace



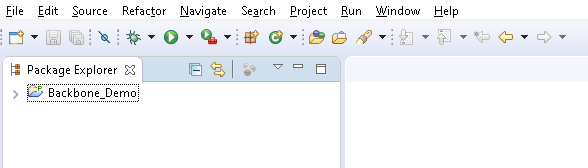
Clickon *Next’*



Select the zip file by clicking on ‘*Browse* ‘button



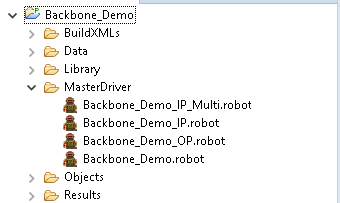
Click on ‘*Finish’* button to import it.



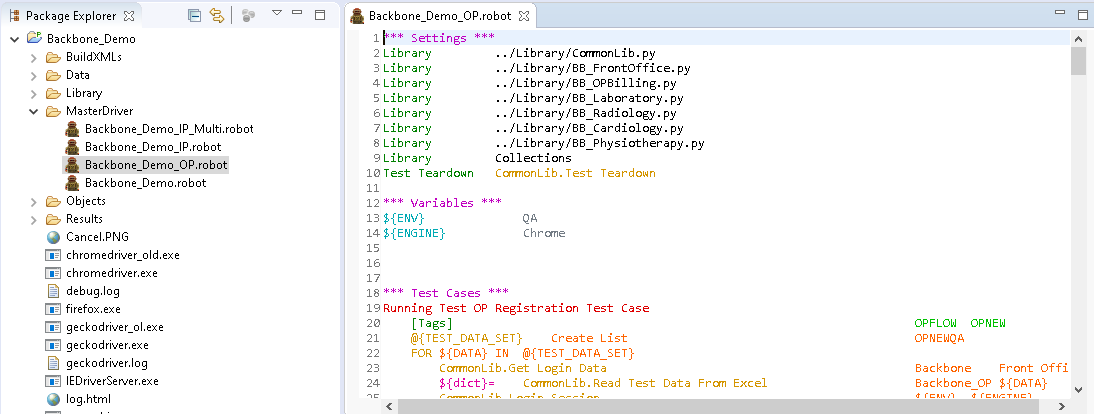
## Preparing to run automation tests

### Settings

#### Expand the Master folder to open the ‘Robot’ file



#### Double Click on the robot file to open it.



#### Now open Run menu -> External Tools -> External Tools Configurations… to update the below run settings

a. Name: *Robot*

b. Location: *C:\Python27\Scripts\robot.exe*

c. Working directory: Path of workspace *e.g. ${workspace\_loc:/Backbone\_Demo}*

d. Arguments: "${selected\_resource\_loc}"

e. Click on Apply

All the above steps are required only for the first time .

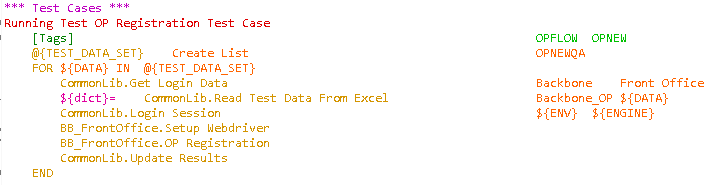
### Controlling run parameters

#### Creating run batches

##### Selecting specific test cases to run for any release:

Tag

Test case name



E.g., “OPFLOW and OPNEW” are the tag names for test case ‘*Running Test OP Registration Test case’*

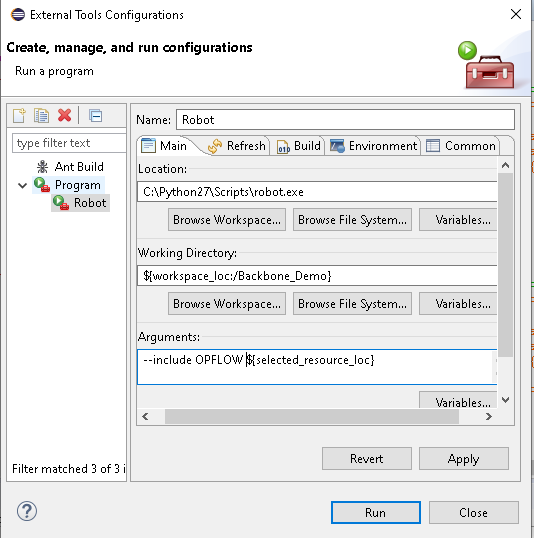
**Steps for running test cases in Batches:**

1. Identify the test cases to be run for Batch1 and Batch 2
2. Change the tag name of Batch 1 test cases in the robot file E.g. ‘OPFLOW*’*
3. Change the tag name of Batch 2 test cases in the robot file E.g. ‘IPFLOW*’*
4. Open eclipse and Open Run menu -> External Tools -> External Tools Configurations
5. Add the *‘--include OPFLOW’* in argument as highlighted in the above screenshot and click on *Apply* and *Run*

Only the test cases with tag name as ‘OPFLOW’ will be run and other test cases will not run.

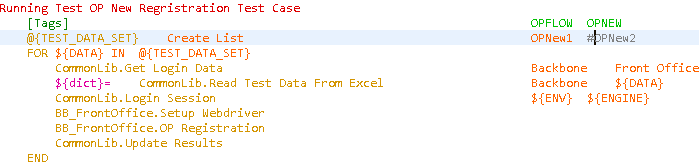
Once the batch1 run is complete, we can change the argument to *‘--include IPFLOW’* and run the batch 2 test cases.

This way execution of automation test cases can be controlled for any release.



#### Running tests for multiple data

#### For running the same test case for multiple data:



Multiple data are inputted as datasets in the robot file as shown in the above screenshot. Each test case will have separate datasets.

E.g. OPNew1*, OPNew2*

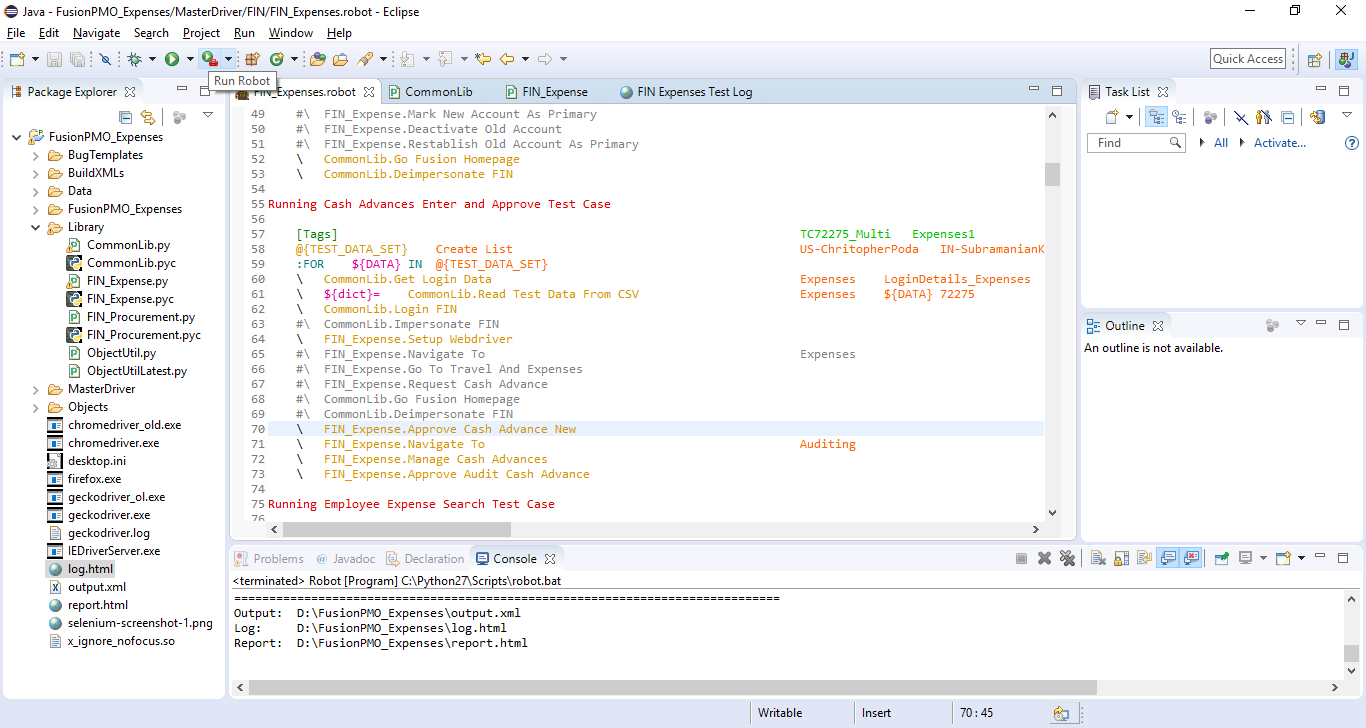
If the name is in red like OPNew1 in the above screenshot means test case will run for this dataset. If it in grey *OPNew2* means it will not run for this dataset.

If it needs to be run for both datasets then we just need to remove the ‘*#’* before the dataset to enable it

Adding a *‘#’* before the dataset will stop selenium from running the test case for that dataset.

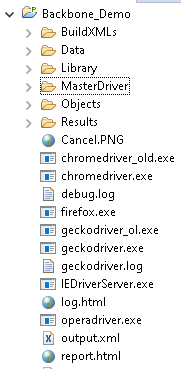
### Running the Automation script

Once the parameters are set and countries are update in the robot file, scripts can run by clicking on the ‘Run Robot’ button as shown below



All scripts will run one by one based on the number of datasets chosen. Once the scripts run is complete a log file, a report file and screenshots (if there are any failures) will be generated.

### Log file and Screenshots



Following reports will be generated at the end of the execution of all scripts.

* log.html
* report.html
* Selenium-screenshot – (number).png - This will be generated only when there are any failures. Each failed test scripts will have a separate screenshot.

**Additional installs:**

#### Open command prompt and change the directory to C:\Python27\Scripts by running the command ‘CD C:\Python27\Scripts’

1. Install openpyxl - used for reading excel and csv files.

- pip install openpyxl

1. Install xlsxwriter - used for writing to excel files

- pip install xlsxwriter

1. Install pyautogui - used to handle non web windows

- pip install pyautogui