Sohail Abbas

leterbag@hotmail.com

May 20, 2020

This document will help with detailing about the automation usage

Automation framework user guide

Version 1.0

Table of Contents

[Introduction to Automation Framework 2](#_Toc41072390)

[Why Automation? 2](#_Toc41072391)

[Why automate using VB Script? 2](#_Toc41072392)

[Automation Framework & Flow 3](#_Toc41072393)

[Components of framework 3](#_Toc41072394)

[Master Script 3](#_Toc41072395)

[Functional Library 4](#_Toc41072396)

[Input File (Test Sheet.xlsx) 4](#_Toc41072397)

[Output: 4](#_Toc41072398)

[Git Clone and Setting up the path: 6](#_Toc41072399)

[How to Create Input Data and Execute Automation? 6](#_Toc41072400)

[Identification of Various Date Format & Creation of Data: 6](#_Toc41072401)

[Execution 7](#_Toc41072402)

# Introduction to Automation Framework

This framework is designed using vbscript to run regression for Propine Date Parser webpage.

## Why Automation?

When we talk about continuous testing in CI/CD pipeline then we are more focused on executing current and old scenarios in a progressive way, hence automation is very well necessary not only to just save time and money but to also increase the efficiency and effectivity of the testing process, hence we can list following benefits of automation here

1. Automation testing increase the efficiency and effectivity as it make regression and continuous testing easy to execute.
2. Less human intervention is required, hence we save lot of money and time in long run.
3. In CI/CD way time means a lot, hence this will speed up test completion process.
4. Better Test Coverage
5. Manual Testing can become boring and hence error-prone.

## Why automate using VB Script?

This automation could have been done in couple of languages or tools like QTP or Selenium, but I didn’t use any as it required to install. Also, there are couple of free framework available in market but I haven’t used any as this was simple webpage validation.

Hence, I preferred VB Script as no tool will be required to install to execute this and it is quite easy and simple to learn and adapt. Please note that I could have used other scripting language as well like JavaScript but have used vb script again due to its simplicity. However, this can complete code can be migrated to JS as or whenever required.

## Automation Framework & Flow

Master Script

Function Library

Test Data.xlsx

**HTML Report**

Excel Report

Reports

VB Script

## Components of framework

**Master Script:** This is the main script segment where the test cases are driven in order to perform various execution cycle.

**Function Library:** This contains vbscript based functions that are frequently used for various actions within the framework.

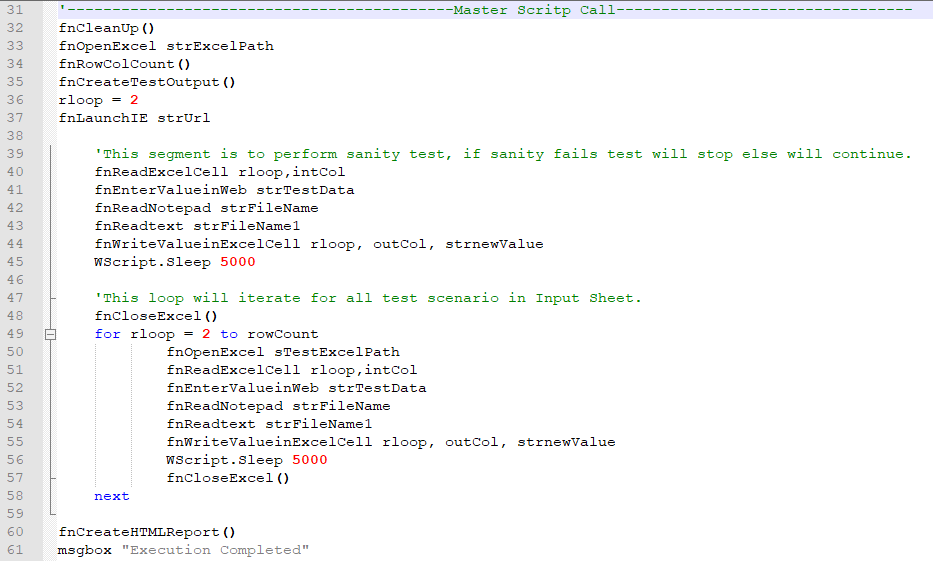
**Template:** This contains report template and propine logo image.

**Test Data file:** This file is the main input file for the test execution. It has all different scenario and the different date format that needs to be inserted in the test cases.

**Reports:** This folder contains excel report and html report which gives detailed reporting for the test cases executed.

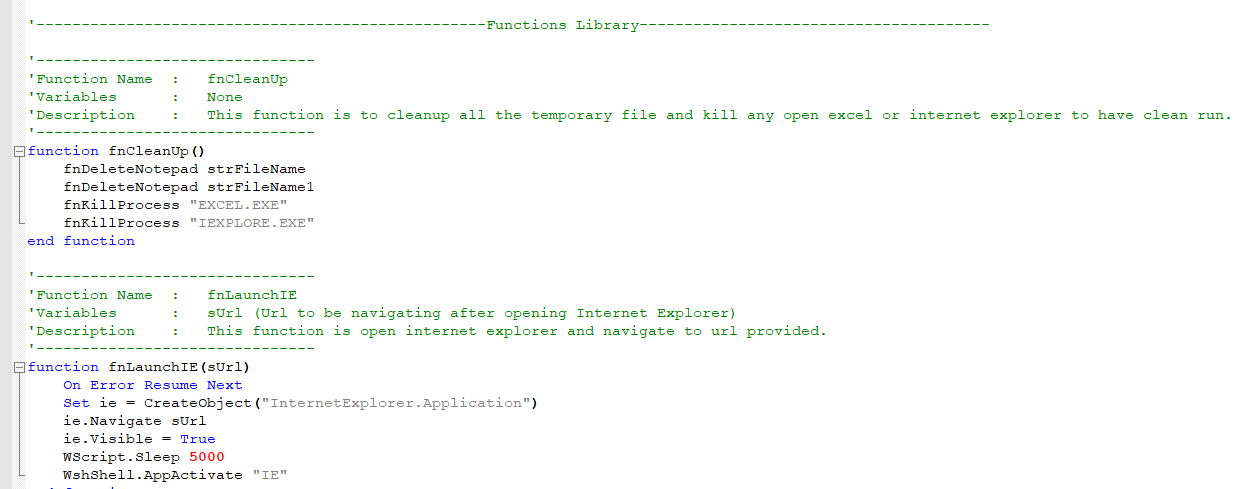
### Master Script

This is the main script which lies in vbscript file right after variable declarations under the segment of Master Script Call.



### Functional Library

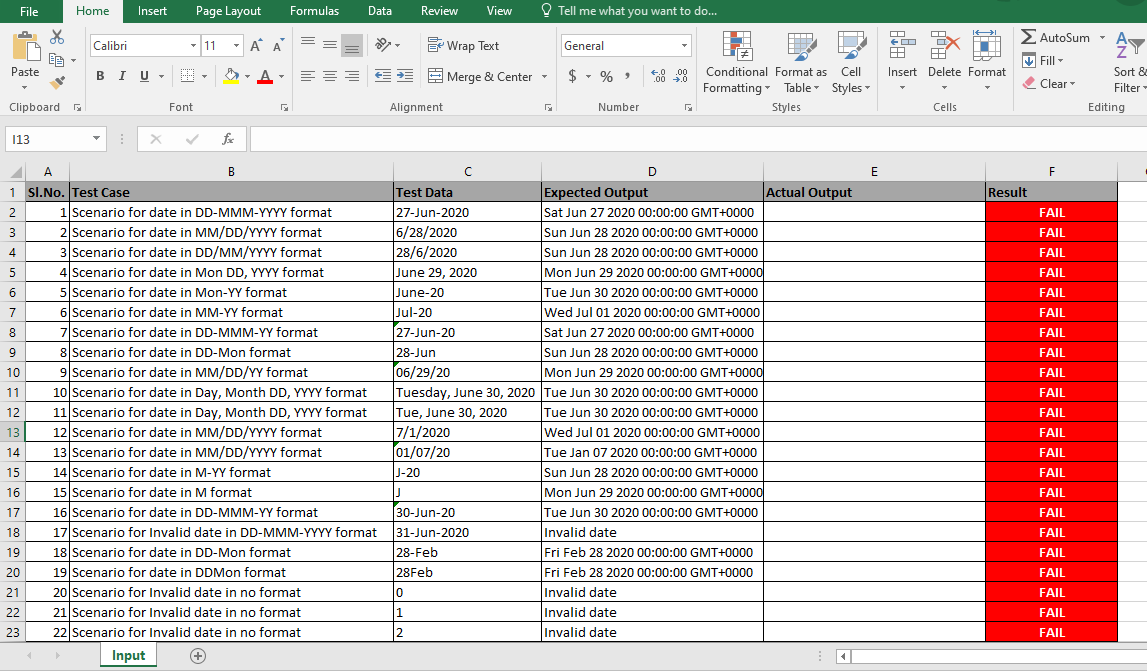
This is the segment of script which lies in vbscript file right after Master Script call and it has all the functions that perform action within the framework.



### Input File (Test Sheet.xlsx)

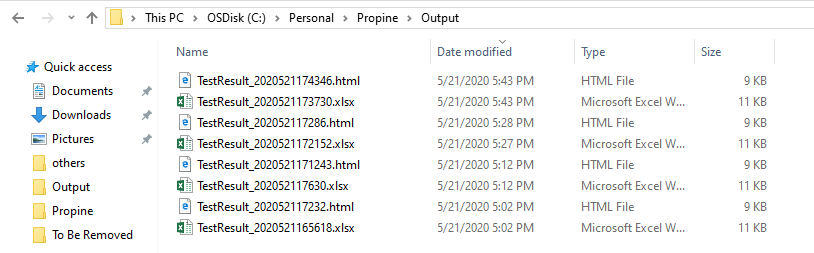
This file contain actual test scenario and test data for the execution and serve as Input file for the automation. This has following columns:

1. Sl. No.
2. Test Case: Name of Test Scenario
3. Test Data: Actual Test Input Data
4. Expected Output
5. Actual Output: Left as blank which will be populated in excel report.
6. Result: This column has formula to compare Expected and Actual and display as Pass or Fail, this is in beginning set to “Fail” as Actual Output field in null. This is also for result sheet.



### Output:

Output folder has our report consist of two files, one is Excel Report and another is HTML Report.



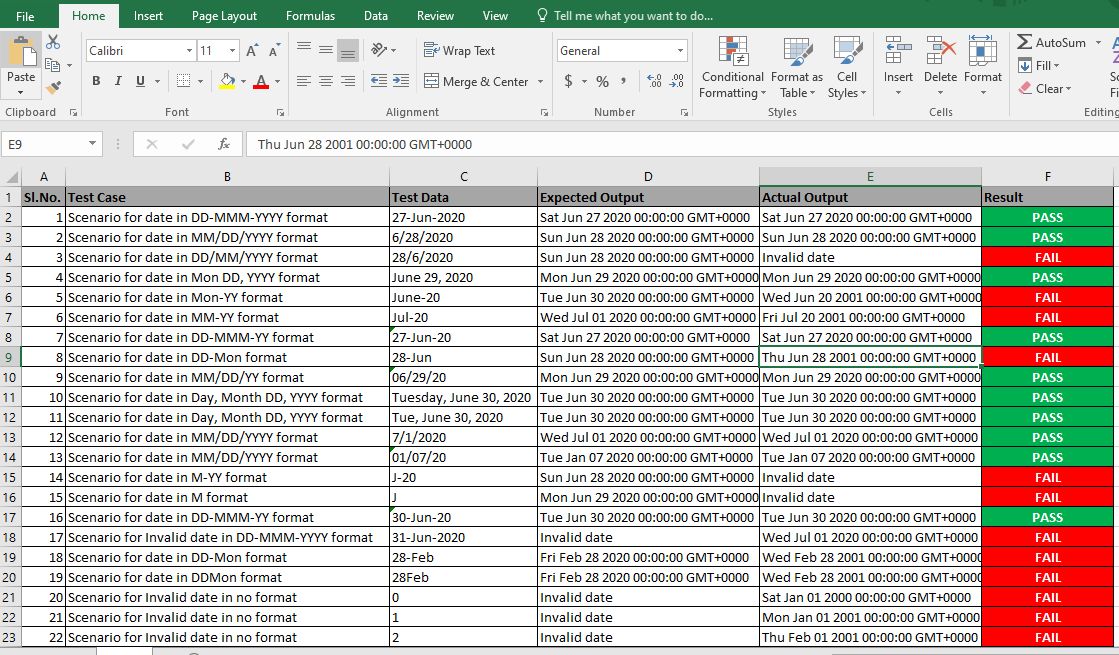
#### Excel Report:

Excel report has similar columns as that of Input Sheet as it is replicated from Input Sheet only, in order to keep history of test execution and keep it separate from the Input Sheet.

This is saved with the name “TestResult\_*DateTimeofExecution*” i.e. like TestResult\_2020521173730.xlsx (This means that this was executed on 21st May 2020 at 5:37:30 PM)

This file also contain actual test scenario and test results for the execution and serve as report:

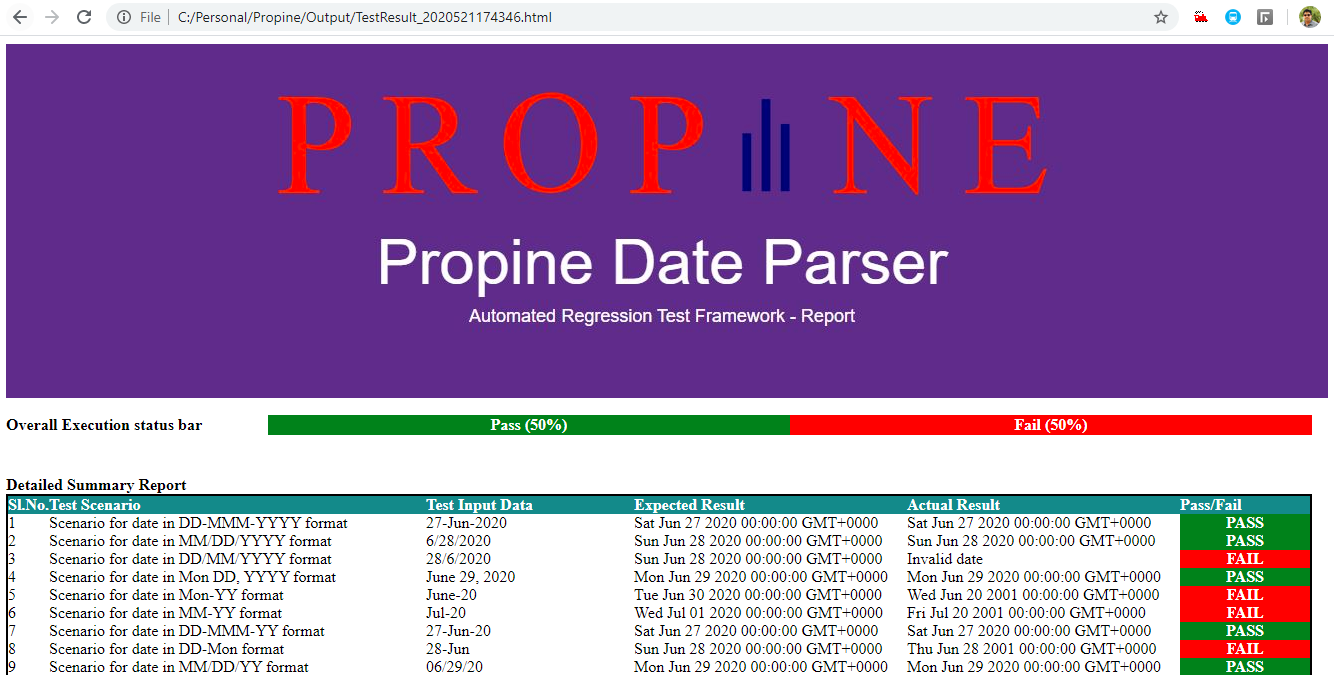
1. Sl. No.
2. Test Case: Name of Test Scenario
3. Test Data: Actual Test Input Data
4. Expected Output
5. Actual Output
6. Result: This column has formula to compare Expected and Actual and display as Pass or Fail.



#### HTML Report:

HTML report is similar as excel report, only addition is that it has the bar of Pass/Fail percentage.

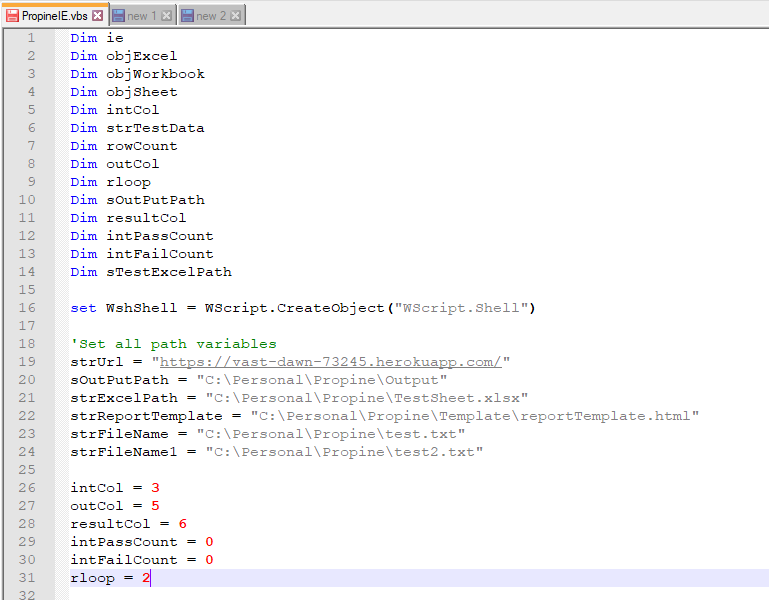
This is also saved with the name “TestResult\_*DateTimeofExecution*” i.e. like TestResult\_2020521174346.html (This means that this was created on 21st May 2020 at 5:43:46 PM)



## Git Clone and Setting up the path:

Git clone following repository in your local: <https://github.com/sohail-abbas/PropineDateParser.git>

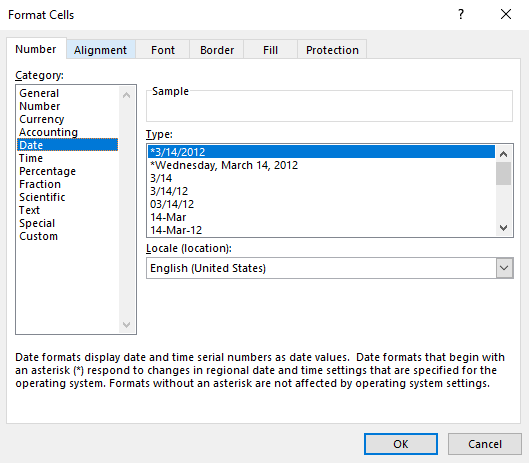
Now, navigate to “PropineIE.vbs” file and change the path where variables are defined.



## How to Create Input Data and Execute Automation?

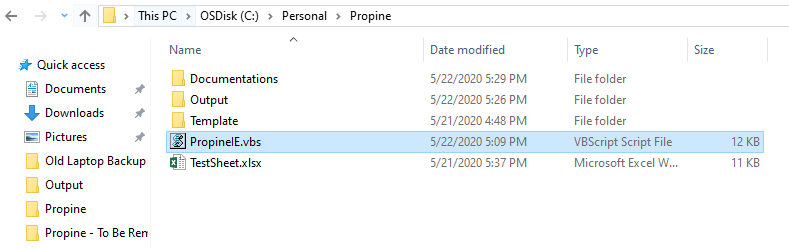
### Identification of Various Date Format & Creation of Data:

To have full coverage of various different date format and add different type of Date format we use the excel feature Format Cell and select Date, we get various types of date format, through this we can add different scenario and its test data to have full coverage:



### Execution

Execution is very easy as we have to just double click on “PropineIE.vbs” file and it will execute complete cycle.



However, before execution please clone the git repository to your local and map the path in the vbscript:

