Client Name

Project Title

DD Mmm YYYY

Version #.#

Type of Document

DocumenT approval

Synchrony IT

|  |  |  |
| --- | --- | --- |
| Name | Position / Title | Sign off Requirements |
| Gary Channer | VP, QA Center of Excellence,  SYF-IT | Review and Sign Off |
| Vinay Gowda | Offshore Automation Stream Lead | Review and Sign Off |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Release Date | Released By | Description of Changes |
| 0.1 | 08.01.2016 | Shivakumar Odogoudra | Draft Version |

Contact Information

If you have any questions regarding this document, please direct your e-mails to:

Gary Channer

Title: VP, QA Center of Excellence, SYF-IT

Company: Synchrony Financial

Email address: Gary.Channer@syf.com

Vinay Gowda

Title: Automation Stream Lead

Company: Attra Infotech Pvt ltd

Email address: [vinay.gowda@syf.com](mailto:vinay.gowda@syf.com)

Shivakumar Odogoudra

Title: Senior Test Engineer

Company: Attra Infotech Pvt ltd

Email address: [shivakumar.odogoudra@syf.com](mailto:shivakumar.odogoudra@syf.com)

Version History of this template

| Version No. | Version Date | Name | Creator / Reviewer / Approver | Description |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Content

[1 INTRODUCTION 5](#_Toc15569410)

[1.1 Objectives 5](#_Toc15569411)

[2 HAF Framework – Design 6](#_Toc15569412)

[2.1 Automation Script Run 6](#_Toc15569413)

[2.2 Main Driver Script 6](#_Toc15569414)

[2.3 Reusable Actions 7](#_Toc15569415)

[2.4 Function Libraries 7](#_Toc15569416)

[2.5 Object Repository 7](#_Toc15569417)

[2.6 Automation Reports 8](#_Toc15569418)

[2.6.1 UFT Default Report 8](#_Toc15569419)

[2.6.2 Excel Report 8](#_Toc15569420)

[2.6.3 ECD Report 9](#_Toc15569421)

[2.6.4 Email Notification 10](#_Toc15569422)

[3 Test Script Execution Workflow 10](#_Toc15569423)

[4 Test Script Preparation Procedure 11](#_Toc15569424)

[5 HP ALM and UFT Structure 12](#_Toc15569425)

[5.1 Development Phase 12](#_Toc15569426)

[5.2 Execution Phase 14](#_Toc15569427)

# INTRODUCTION

Automation testing is an emerging field that draws maximum benefits with minimum effort. Then benefit of automation testing is its ability to increase the efficiency of resources, increase test coverage, and increase the quality and reliability of the software.

While there are several frameworks that provide support for automated software testing, this document introduces one particularly effective type: Hybrid Test Automation Framework.

## Objectives

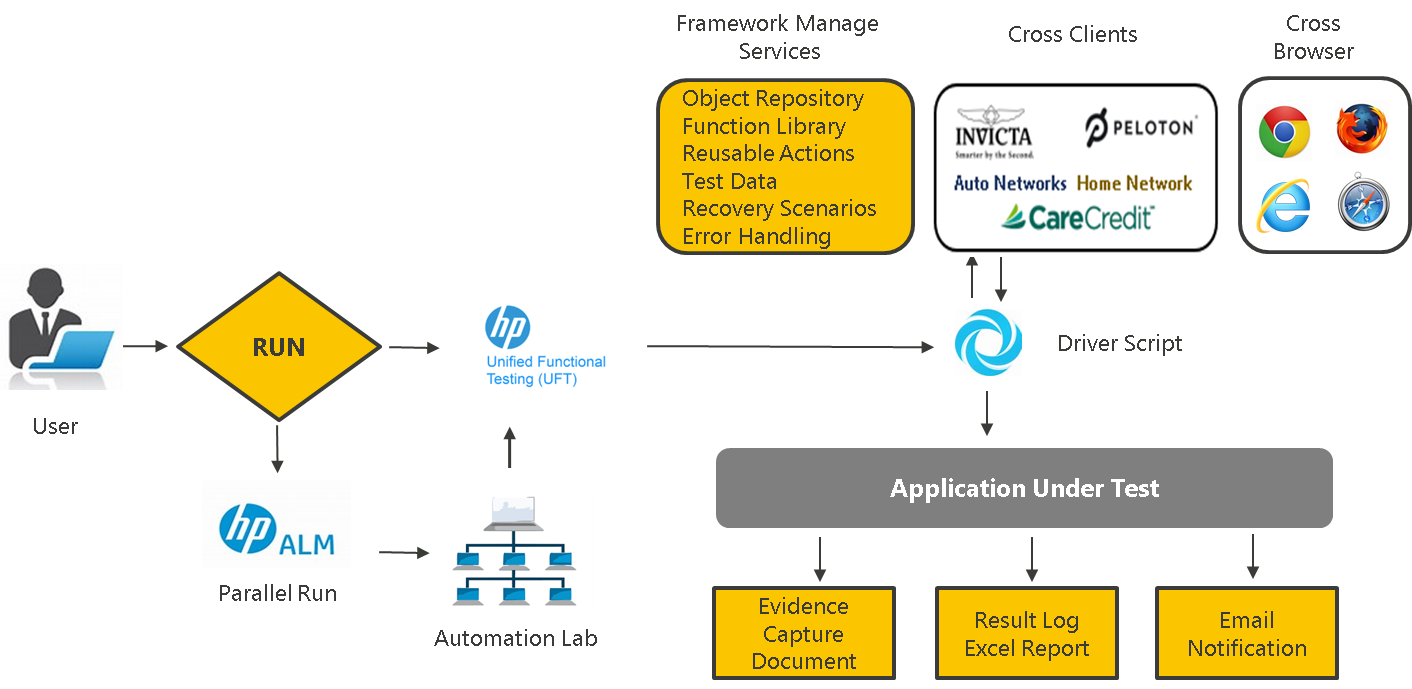
Strategic objectives in building the below framework would be addressed as detailed below

* Implement a strategy that will allow tests to be developed and executed both manually (initial test cycle) and via an automation framework (regression test cycles).
* Separate test design and test implementation to allow test designers to concentrate on developing test requirements, test planning, and test case design while test implementers build and execute test scripts.
* Implement a testing framework that both technical and nontechnical testers can use.
* Employ a test strategy that assures that test cases include the navigation and execution steps to perform, the input data to use, and the expected results all in one row or record of the input data source.
* Realize an integrated approach that applies the best features of keyword-driven testing, data-driven testing, and functional decomposition testing.
* Implement an application-independent test automation framework.
* Document and publish the framework.
* Develop automated build validation (smoke) tests for each release of the application.
* Develop automated environmental setup utility scripts for each release of the application.
* Develop automated regression tests for
  + GUI objects and events Application functions
  + Application special feature
  + Application performance and scalability
  + Application reliability
  + Application compatibility
  + Application performance
  + Database verification

# HAF Framework – Design

Below mentioned diagram depicts the automation test execution workflow designed using the Hybrid Automation Framework (HAF).

Exhibit Hybrid Automation Framework – Architecture



## Automation Script Run

User Prepares the Test data and selects one of the Execution modes

* **HP ALM Integration mode** - HP ALM Opens HP UFT application using HP ALM - UFT Add-in and triggers the automation scripts based on the test execution flag “Y” or “N” from the test data sheet placed in the Resources module.
* **Standalone mode** - User Opens UFT application through desktop and navigate to the automation framework folder structure to select the specific driver script which in turn triggers the automation scripts based on the test execution flag “Y” or “N” from the test data sheet imported to the data table when clicked on “Run” button.

## Main Driver Script

* Driver script fetches the framework folder path from the environment variables which are inbuilt associated and associates global function libraries which are used to generate excel reports.
* Driver script loads the test data to the data table.
* Once the test data is loaded based on the keyword selection, driver script calls the reusable actions.
* Driver script will also generate the test execution result based on the test execution status returned by the reusable action (E.g.: Pass/Fail/Timestamp etc.…)

## Reusable Actions

* Reusable actions will associate the required function libraries and loads the runtime environment variables which include application links
* Reusable actions will load page specific object repositories depending on the keywords.
* Required parameter for reusable actions are passed from driver scripts
* Based on the input parameters reusable actions performs operation on the AUT (application under test) and updates the test result summary and status back to driver script
* Reusable actions also generate HP UFT Run Result viewer summary report.
* Reusable actions are built at application page level and at times it also acts as individual keyword.
* Same reusable actions are being used by multiple keywords which brings reusability feature of the automation framework.
* Reusable actions are built with unique naming convention which has application name as prefix followed by Page Name (E.g.: Application: Care Credit Page Name: Login, Reusable Action name = CCLogin)
* Parameters required for reusable actions are provided through test data sheet which are given in the below mentioned columns
* StepNumber
* ActionDesc
* ResultExpected
* KeywordID
* RowCount
* NegativeTest

## Function Libraries

* Function Library (FL) forms the backbone of the Automation Framework.
* All the coding logic is in the form of a user-defined VB script. All these functions are stored in the FL.
* It is the place where most of the scripts reside and the place where customization can be done in the script for the project.
* The FL is the only component in the framework that must be changed in case the application is migrating from one platform to the other. This addition and deletion of functions makes the framework flexible enough to use it for any other application.

## Object Repository

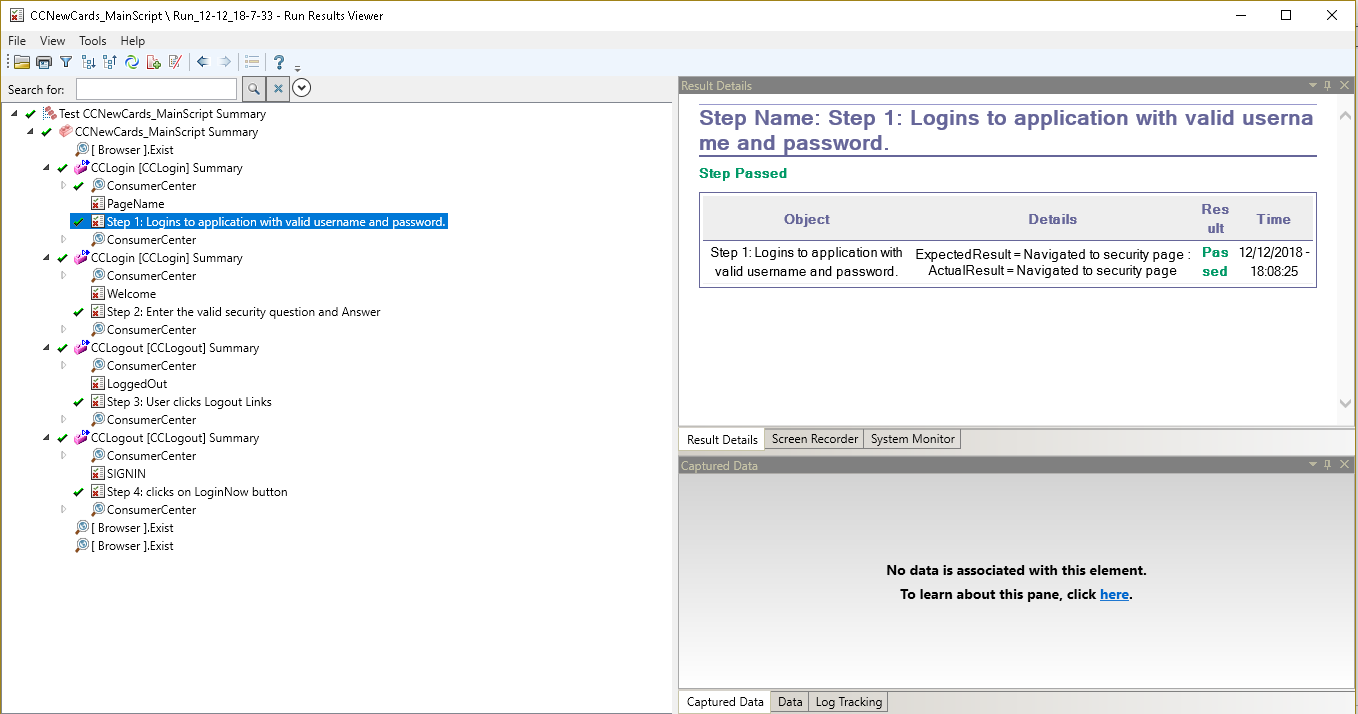
* Application object properties specific to each page are maintained separately using Page Object Model (PoM) approach.
* Page wise object repositories (ORs) are maintained for every reusable action
* PoM approach allows the existing automation framework easy to maintain as any object properties are changed, it can be updated in specific page object repository.
* PoM approach will avoid duplication of object creation compared to test scripts wise object repository.
* PoM approach will also help in reducing test script build.

## Automation Reports

* Automation test summary report generated after every test execution consists of test case wise summary (E.g.: Pass/Fail) along with the timestamp
* HP UFT Run Result viewer contains test step level details with actual and expected results updated

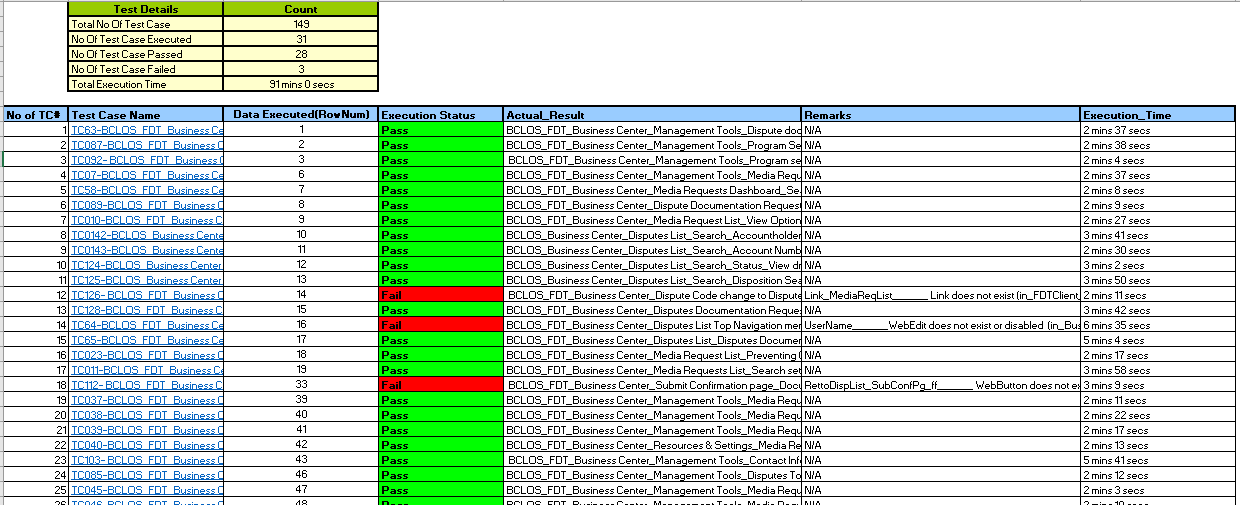
### UFT Default Report

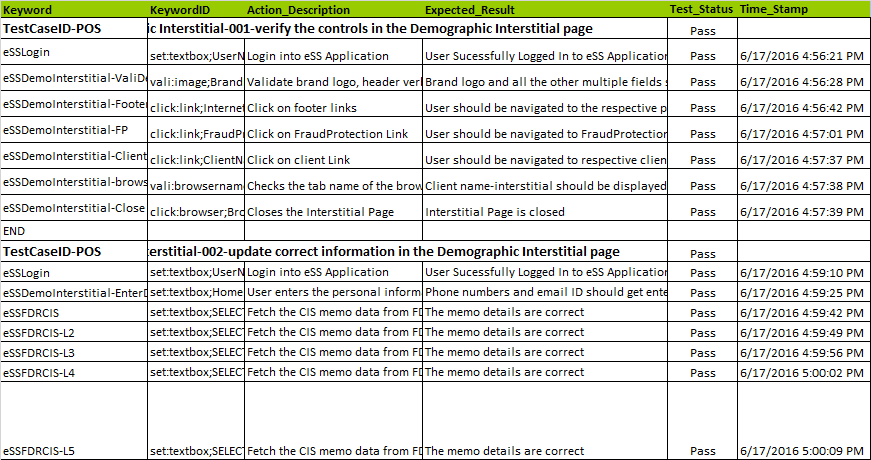
* UFT generates default run results depending on the reporter events passed from the test scripts, below is the sample screen shot for the same,



### Excel Report

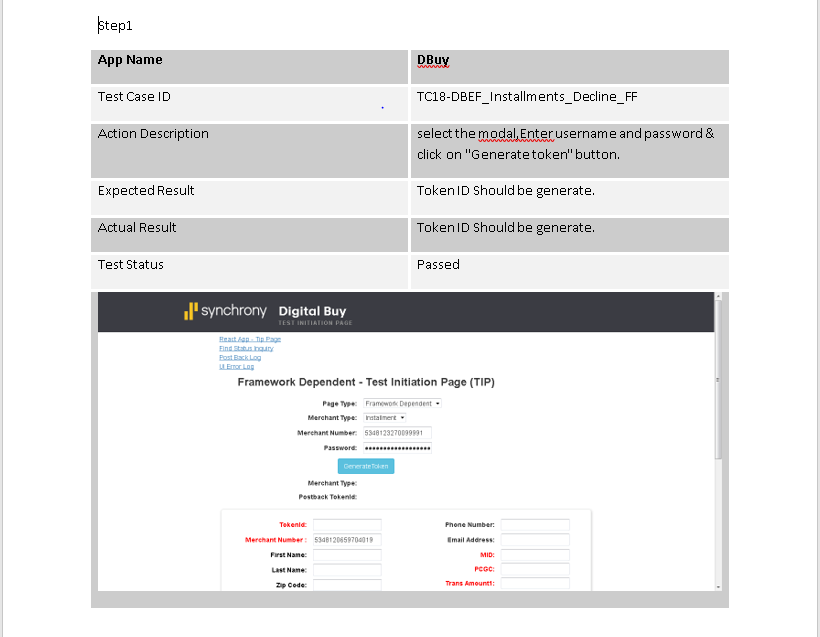
* Excel reports are the customised report which provides the summary of the test run, below is the sample screen shot for the same,





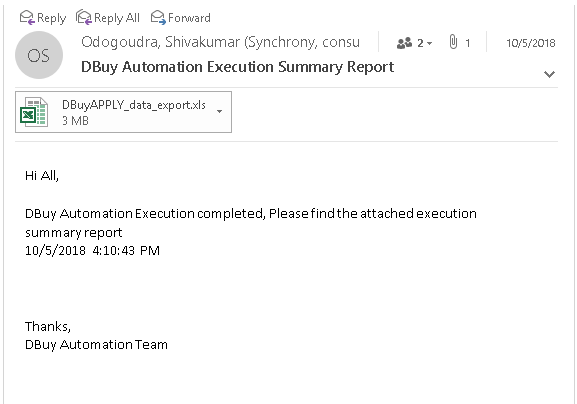
### ECD Report

* Evidence capture document is a word document which captures step by step details of the automation execution with screen shots,

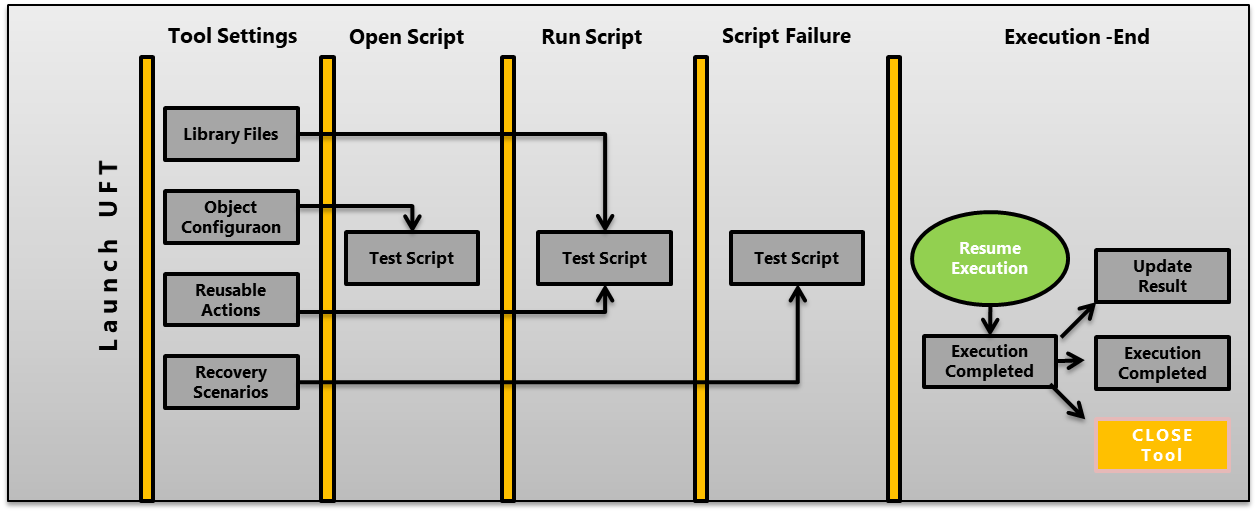


### Email Notification

* Email will be triggered once the execution is completed. The email contains the excel complete report and some status message.



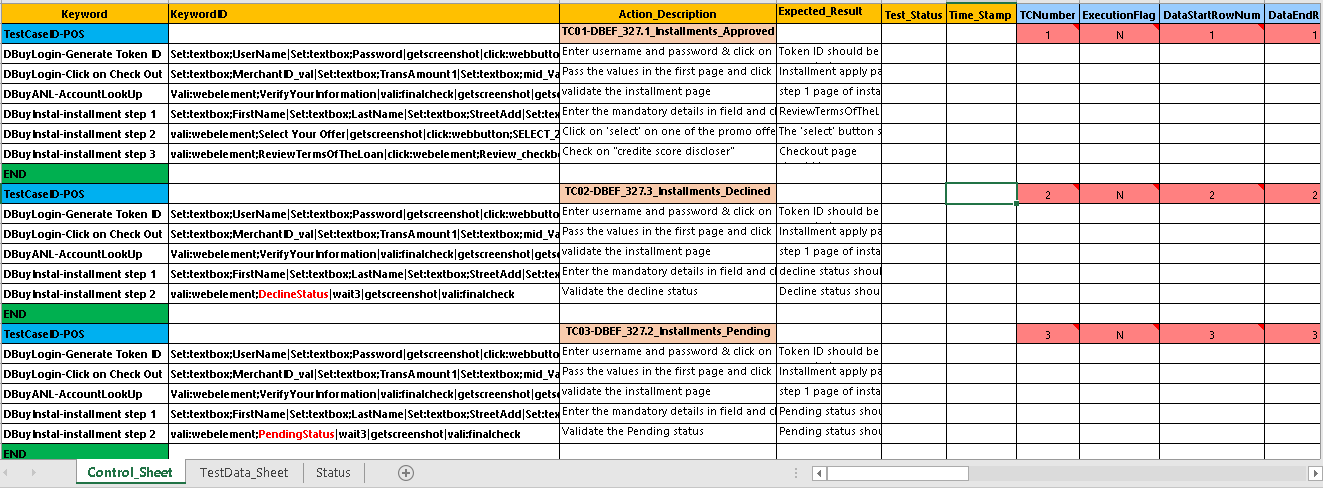
# Test Script Execution Workflow

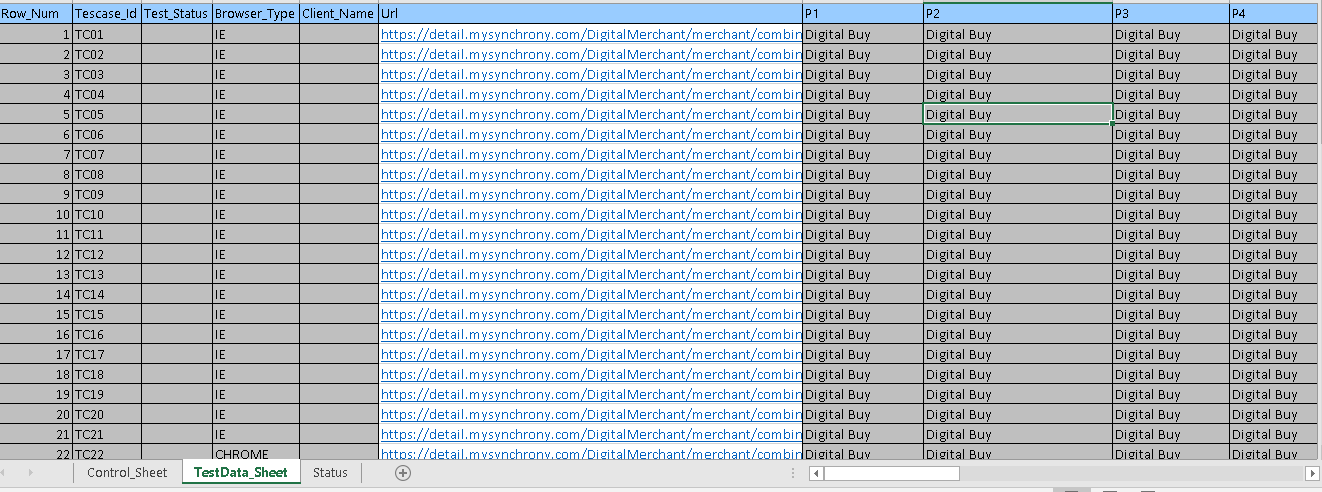
****

* **Tool:** UFT Tool which contains framework managing services like Library files, Object Repository, Reusable actions, Recovery Scenarios will be launched.
* **Scripts:** This folder contains local test scripts; this folder will be used while development and maintenance stage. Once script gets delivered it will upload to centralize location e.g.: Test Director or ALM.
* **Recovery scenario:** This folder can be used as recovery scenario function library file. Also, through Automation tool scenario, converted file is kept under this folder and associates the test script.
* **Results:** The current execution status will be updated after every run. All the Script failures, test status, recovery actions will be logged in Result; ECD’s (Evidence Capture Document) would be generated and updates the status and attach ECD’s to test management tool (ALM).

# Test Script Preparation Procedure

Test scripts will be written in ‘**Control\_Sheet’** and test data will be given in **‘TestData\_Sheet’.** Test data will be take by the test script based on the Start and End row specified in the Control\_Sheet.



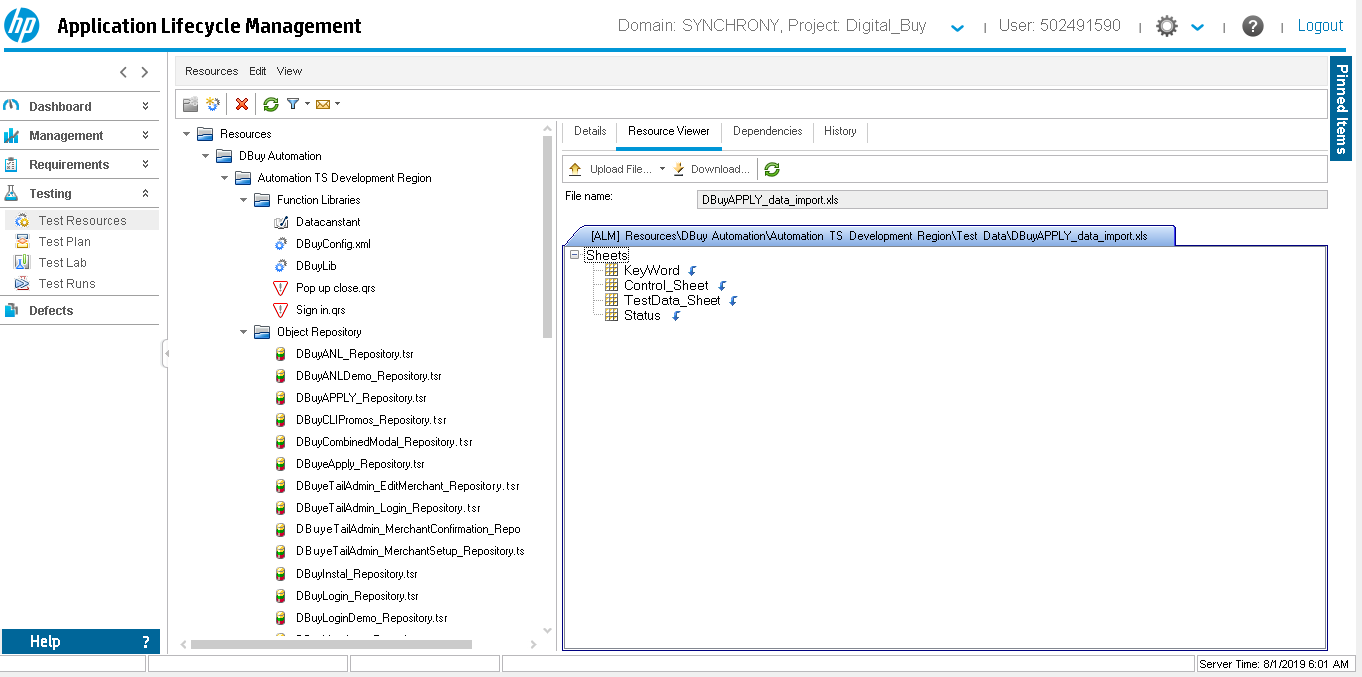


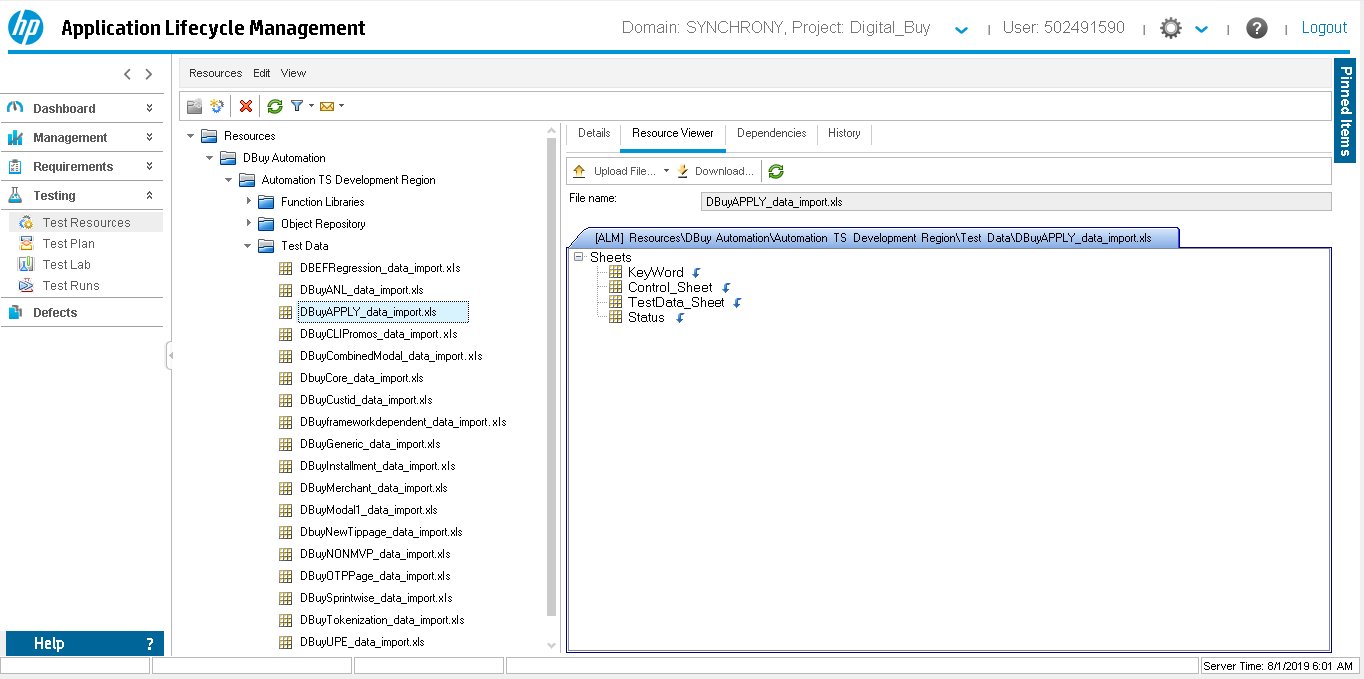
# HP ALM and UFT Structure

## Development Phase

**Test Resource:** Below files are stored in Test Resources Module under respective folder

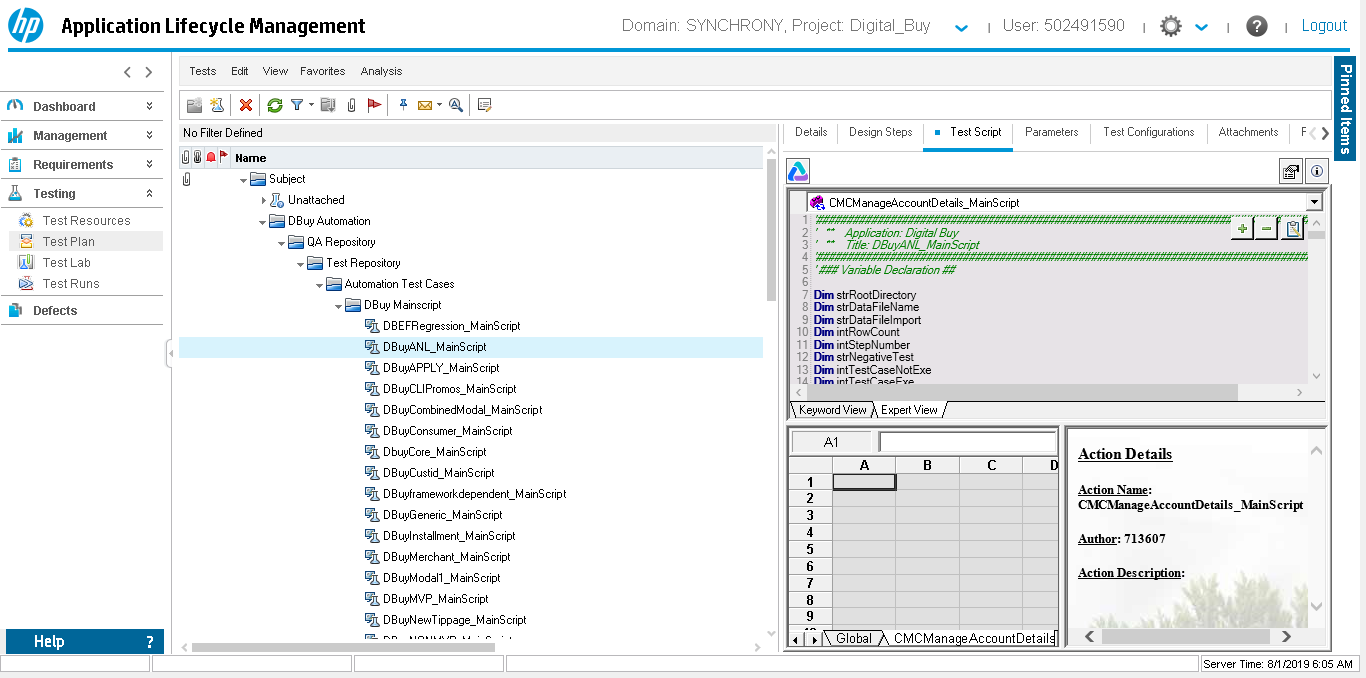
* Test Data
* Function Libraries
* Object Repositories

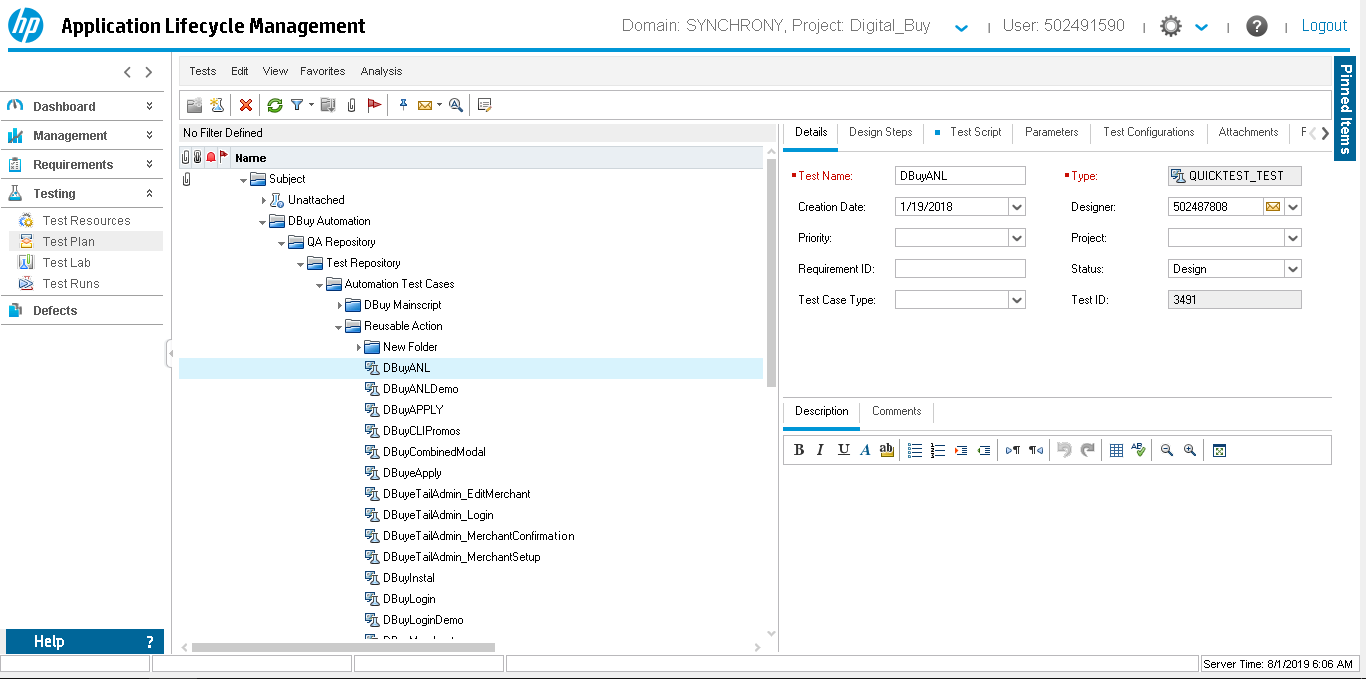




**Test Plan:** Below files are stored in Test Plan Module

* Main script files (Module wise)
* Reusable actions (Combined)

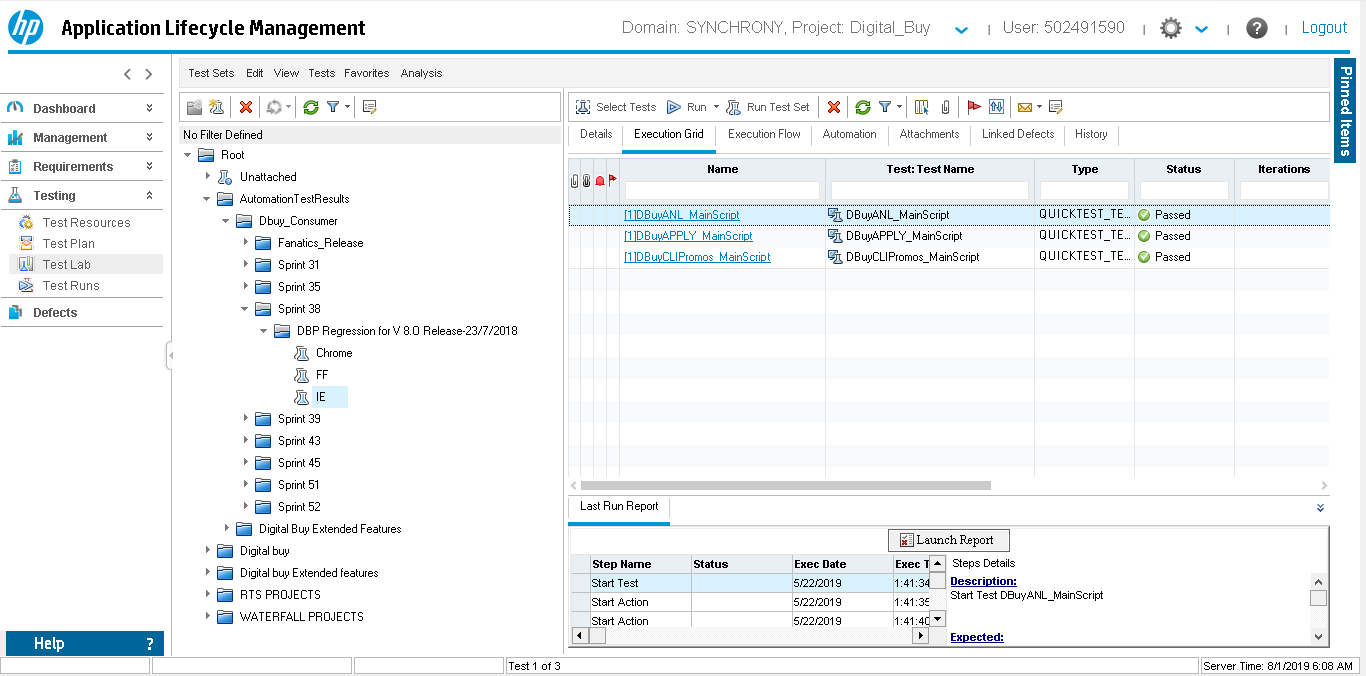




## Execution Phase

**Test Lab:** Create Folder Structure Same Test Plan and move Main Scripts

* Test Set (module wise for 3 browsers)



**Test Runs:** Test results will be available in the test runs along with the excel report.

