|  |  |  |  |
| --- | --- | --- | --- |
| **MICHELIN** | Author : **CGI** | Edition date :  **08-JUNE-2017** | Page **1/6** |
| Réf : | Project : **ENGAGE PROJECT-AUTOMATION** | **FRAMEWORK**  **DESIGN** | Classification : **D3** |

# Automation Framework Design Document

|  |
| --- |
| PURPOSE |
| The purpose of this Documentation is to provide insight to Framework Design for all stake-holders to understand and implement the test automation.   |  |  | | --- | --- | |  |  | | SCOPE | | | Automation testing scope of Engage-project includes   * Framework structure. * Framework guidelines for Engage project-automation for different phases of testing. * Instructions for Framework usage.  HISTORY OF THE DOCUMENT    |  |  |  |  |  | | --- | --- | --- | --- | --- | | Version | Edition Date | Author | Modified Page | Comments | | 1.0 | 8th June 17 | **CGI** | ALL | Framework design for automation | |  |  |  |  |  | | | |  |  | |

**1. FRAMEWORK OVERVIEW:**

Testing automation framework is an execution environment for automated tests. It is defined as the set of assumptions, concepts, and practices that constitute a work platform or support for automated testing.

The Testing framework is responsible for:

* Defining the format in which to express expectations.
* Creating a mechanism to hook into or drive the application under test
* Executing the tests
* Reporting results

Properties of a testing framework:

* It is application independent.
* It is easy to expand, maintain and perpetuate.

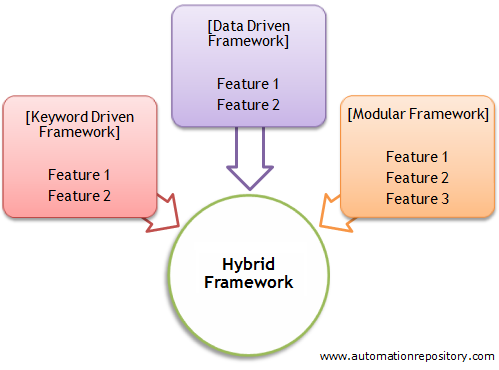
Benefits:

* Helps in avoiding duplication of test cases automated across the application.
* Helps teams organize their test suites and in turn help improve the efficiency of testing.

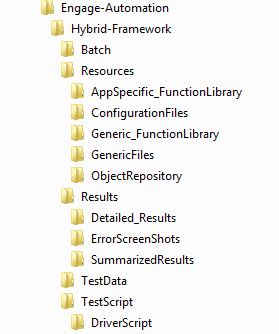
**2. HYBRID FRAMEWORK:**

Hybrid framework is the most commonly used framework in test automation projects. A hybrid framework is a framework that is created by combining together the features of the different types of QTP frameworks.

1. Hybrid framework is a collection of features from the other framework types.
2. A hybrid framework is one that suites our requirements.

****

## 2.1. Framework structure



### 2.1.1. Batch Folder : This folder takes care of the batch execution. The batch excel sheet which will have the list of all the test cases

### 2.1.2. Resources : This is just a top-level folder that has many sub-folders inside it.

**2.1.2.1. AppSpecific\_FunctionalLibrary :** Application specific functions will be stored under this folder. Example : Bill To Account Creation, Ship To Account Creation

**2.1.2.2. ConfigurationFiles :** This folder contains the items that is used to configure the overall framework settings. For example, the settings such as whether to send email or not, or what should be the default wait time etc will be stored in this folder.

**2.1.2.3. Generic\_FunctionalLibrary :** Common Functions will be stored under this folder. Example : Login, Logout

**2.1.2.4. GenericFiles :** This folder is used to store the generic files. For example, if your test cases upload or download some files to/from the application, then those files will be stored in this folder.

**2.1.2.5. ObjectRepository :** This folder contains object repositories for the scripts.

**2.1.3. Results :** This folder contains the following sub folders :

**2.1.3.1. Detailed\_Results :** This folder contains results of the scripts executed.

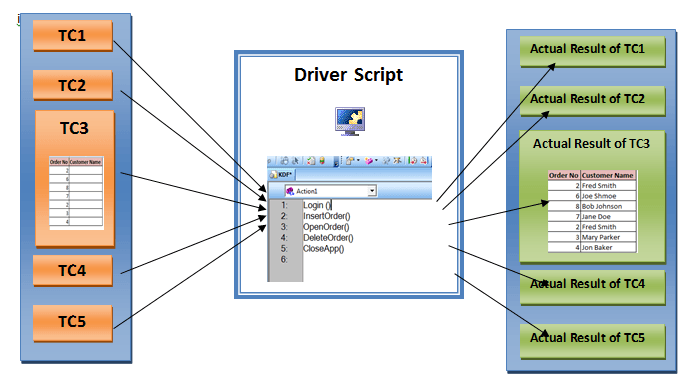
**2.1.3.2. ErrorScreenShots :** This folder contains the error screenshots during execution.

**2.1.3.3. SummarizedResults :**

**2.1.4. TestData :**  All the excel TestData sheets will be stored in this folder

**2.1.5. TestScript :** In this folder all the test Scripts will be stored.

**2.1.5.1. DriverScript :**  This script is used to call the business scenerios test scripts under different phases of testing.



**3.Framework Guidelines :**

1. Data should not be hard-coded in the script.

* The paths of the scrpits and test data should not be hard-coded
* Data should be passed from an excel sheet and place in respective fields

Syntax:

datatable.AddSheet <NewSheetName>  
Datatable.ImportSheet <DataSheet Filepath>,<SourceSheetName>,<NewSheetName>

* To intialize the values from Excel sheet we use the below syntax

BusinessUnit  = datatable.Value(“<ColumnName>”,"Sheet1")

1. Test results should be logged into a separate log file and should be fairly easy to understand.

**4 .Coding Standards :**

**4.1. Script header :** Every script should follow the below script header .

'-------------------------------------------------------------------------------------

'ScriptName - Account Creation

'Description - Script executes the account creation flow

'Created By -

'Created On -

'Modified By -

'Modified On -

'Author - CGI

'------------------------------------------------------------------------------------

**4.2. Inline Documentation :** Proper documentation should be done to understand the script.

Ex : To locate different fields in the script following comments are used :

‘Incrementing BilltoAccName

‘Incrementing ShiptoAccName

‘Channel

‘CommercialRelation

**4.3. Naming Conventions :** Unique naming conventions that is easily relatable should be used through out the script

Ex : BilltoAccName

ShiptoAccName

Channel

CommercialRelation

**5. Error handling info:** Error handling features are implemented as part of the framework. Following

**Intialization:** Closing all the open browser instances as soon as the script

execution is invoked.

**TearDown :** Closing all the open browser instances as soon as script

execution is completed.

**Script Specific :** Identifying thecommon failures in the script execution and

handling them appropriately

EX**:** We were able to identify intermittent issues with login in to the portal. The.

error handling was done in such a way that if the login failed once, the script

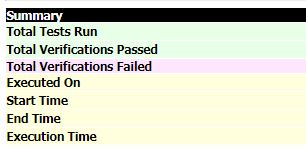
would attempt to login once more.

**6. Result Report :** After executing the script, an HTML Report is created showing the below details, The Result Report should follow the standard.

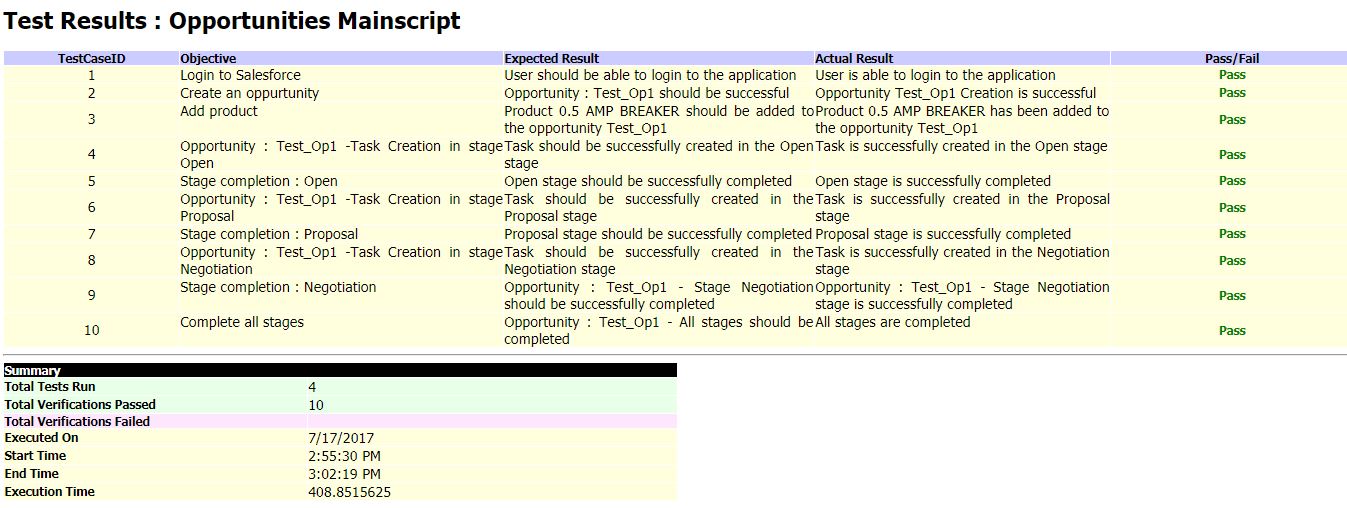
**Given Below is the Report format:**





****

Ex**:** Givenbelowis the Example for Report format



**Path : Hybrid-Framework\Results\Archive,** is the path where Reports are stored

**Syntax:** 1**.** OpenFile "C:\Test.html" , it will open the Result File and will write the

header.

2**.** AddNewCase 1,"Load Homepage","Homepage rendered OK","Page

Rendered","Pass", It will insert a row to validate the checkpoint

3. CloseFile , It will write the summary, and close the file

**7. Variable declarations:** The proper variable decleration should be followed

**Script variables:** Propervariables names should be used with respect to the fileds

inthe application.

Ex**:** OrderSearch

TireSearch

**8. Data sheet info:** Data is passed to the script throuhg an Excel Sheet. The Excel

Sheet should parameters should follow the below standard.

|  |  |  |  |
| --- | --- | --- | --- |
| Run | Paramater1 | Parameter2 | Parameter3 |
| Yes | Value1 | Value2 | Value3 |
| No | Value1 | Value2 | Value3 |

The script will get executed based on the ‘‘Run ‘’ Parameter, if the ‘Run ‘ parameter is given as ‘Yes’ then that specified values will drive the test execution.

**Path : Hybrid-Framework\TestData,** isthe path where the Excel Sheets are stored which drives the test execution

**Github url :** https://github.com/manujohn-bib/AllDeliverables.git

# Reference documents

|  |  |
| --- | --- |
| **Title** | **Document Reference** |
| FDG\_P000007067\_OPE\_Step2\_Interzone\_SDS\_Order Entry | 1.0 |