2014

Cigniti Script less – Test Automation User Manual

Srinivas Gajre

Cigniti Inc

4/25/2014

DOCUMENT HISTORY

Document Version #	Date	Prepared/ Changed By	Approved By	Description
1.0	25-Apr-14	Srinivas Gajre		Initial document

TABLE OF CONTENTS

1. Introduction

The purpose of this document is to outline the usage of the Test Automation Framework designed for the AUT applications.

2. Prerequisites

Software	Purpose
QTP 11.0	Test Automation tool
MS Excel 2007 or above	To create Components, test cases, test suites and maintain the test data
.Net framework 3.5 and above	To register the solution DLL

3. Test Automation Framework

3.1 Framework Overview

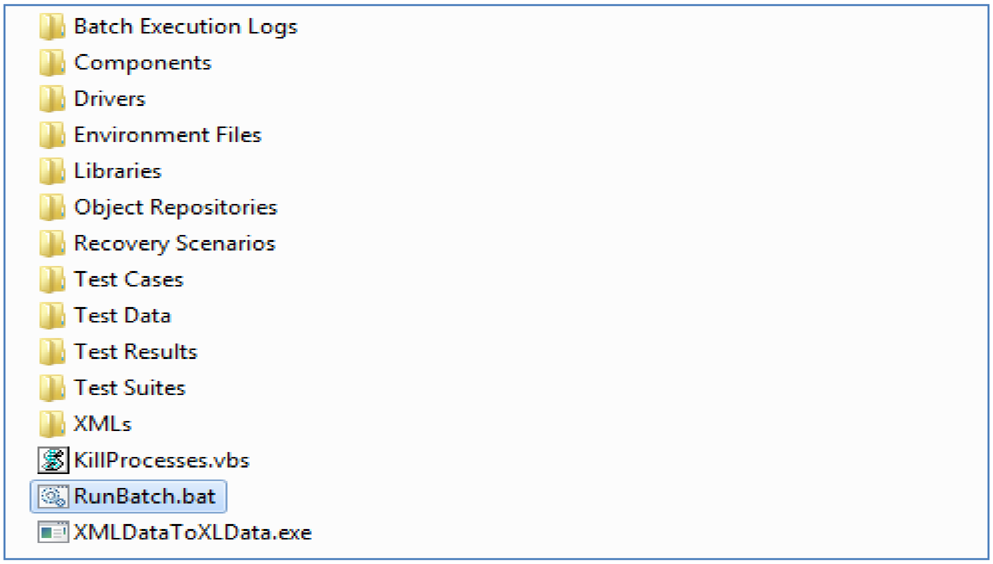
Please refer the Test Automation Framework document

3.2 Framework Architecture

Please refer the Test Automation Framework document

3.3 Framework Folder Structure

Following is the snap shot of the framework folder structure, segregated with the respective folder categories.



Batch Execution Logs
Components
Drivers
Environment Files
Libraries
Object Repositories
Recovery Scenarios
Test Cases
Test Data
Test Results
Test Suites
XMLs
KillProcesses.vbs
RunBatch.bat
XMLDataToXLData.exe

3.4 Customization Overview, Roles and Responsibilities

3.4.1 Customization Overview

1. The framework is portable and just requires copying the framework folder to any of the drives/ folders of the any machine.
2. Doesn't require any specific/ special configurations.

3.4.2 Roles and Responsibilities

1. Only two roles:
   a. Automation Engineer
   b. Subject Matter Expert (SME)

Automation Engineer:

1. Create Object Repositories.

Note: It is always advised to create individual module wise Shared Object Repositories and name them as per modules in the AUT. This way it will be ease the maintenance and be easier while designing the flows, for respective users working on individual or separate modules. Refer section 4.1.

Subject Matter Expert (SME):

1. Create a Manual Component using the Component template provided in the framework. Refer section 4.2.
2. Design the Test steps, by selecting the dynamic drop downs populated with the selection options based on the dependant previous options.

3. Click on the “Generate Component Code & Test Data File” button in the component file, to convert manual steps to automated steps.
4. Create Test Cases using the manual components already created, with test data file name as per test flow. Refer section 4.3.
5. Create a Test Suite with the test cases, to be executed as part of it. Refer section 4.4.
6. Update the Test Data file with the required test data. Refer section 4.5.
7. Run the Test Suite by double clicking the “RunBatch.bat” file.
8. View the Test Results post execution completion in the HTML format.

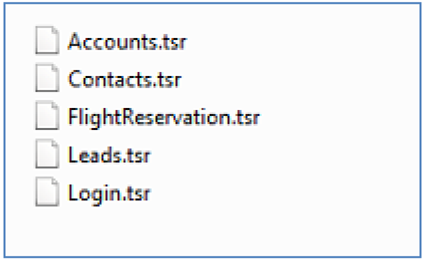
4. Procedure to use the framework

4.1 Object Repository

4.1.1 Folder Path in Framework

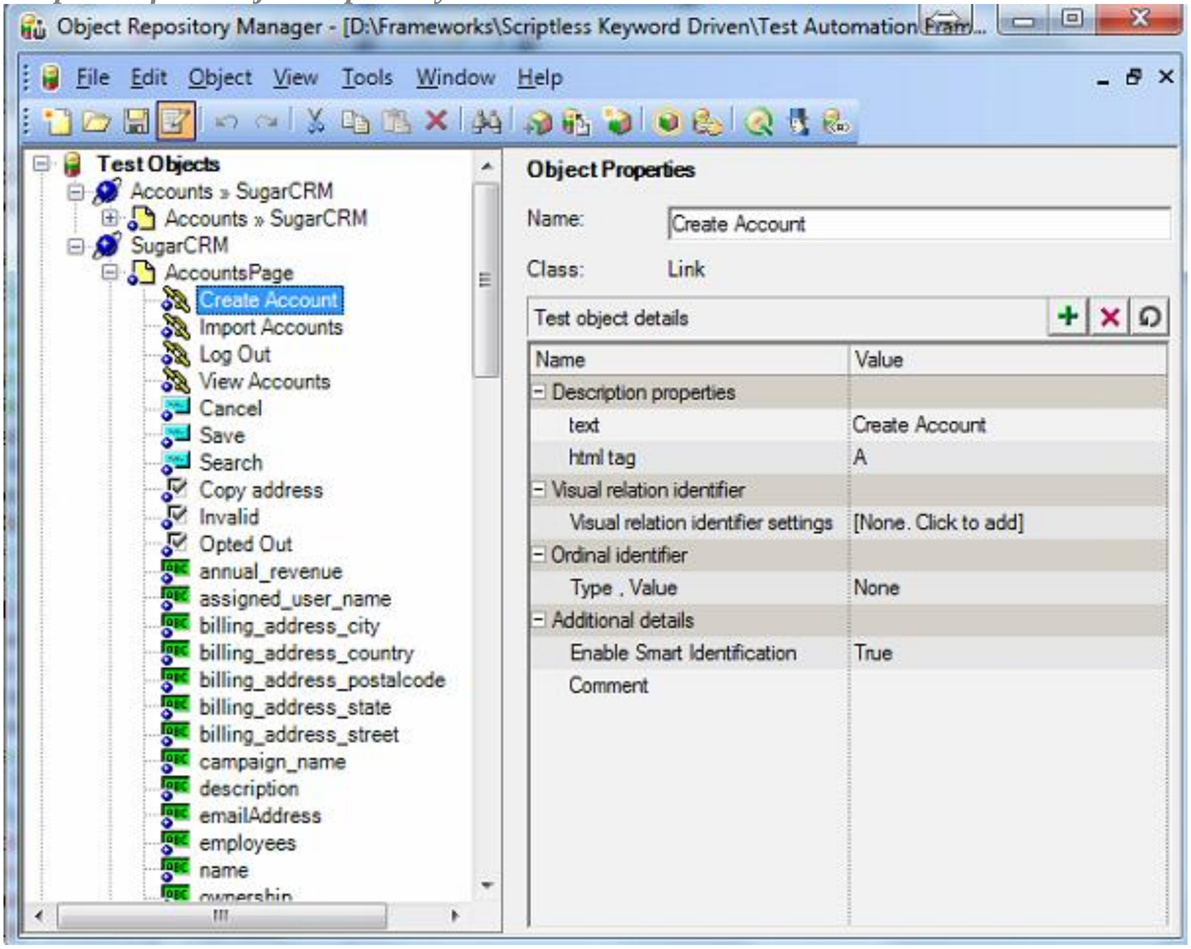
- Test Automation Framework\Object Repositories

Snapshot of the object repositories under Object Repositories folder:



Accounts.tsr
Contacts.tsr
FlightReservation.tsr
Leads.tsr
Login.tsr

Snapshot of the Object Repository:


Object Repository Manager - [D:\Frameworks\Scriptless Keyword Driven\Test Automation Fram...
File Edit Object View Tools Window Help

- Test Objects
  - Accounts » SugarCRM
    - Accounts » SugarCRM
  - SugarCRM
    - AccountsPage
      - Create Account
      - Import Accounts
      - Log Out
      - View Accounts
      - Cancel
      - Save
      - Search
      - Copy address
      - Invalid
      - Opted Out
      - annual_revenue
      - assigned_user_name
      - billing_address_city
      - billing_address_country
      - billing_address_postalcode
      - billing_address_state
      - billing_address_street
      - campaign_name
      - description
      - emailAddress
      - employees
      - name
      - ownership

Object Properties

Name: Create Account

Class: Link

Test object details

Name	Value
Description properties	
text	Create Account
html tag	A
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

Steps to create Object repository:

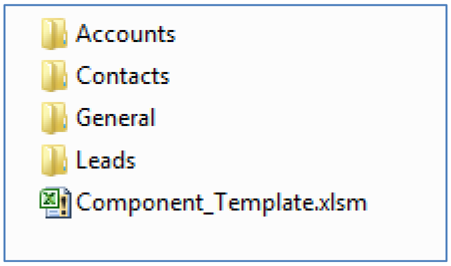
1. Object repository is created using HP QTP tool following the standard method. Refer the HP QTP documentation for more information on how to create the object repository.

4.2 Components

4.2.1 Folder Path in Framework

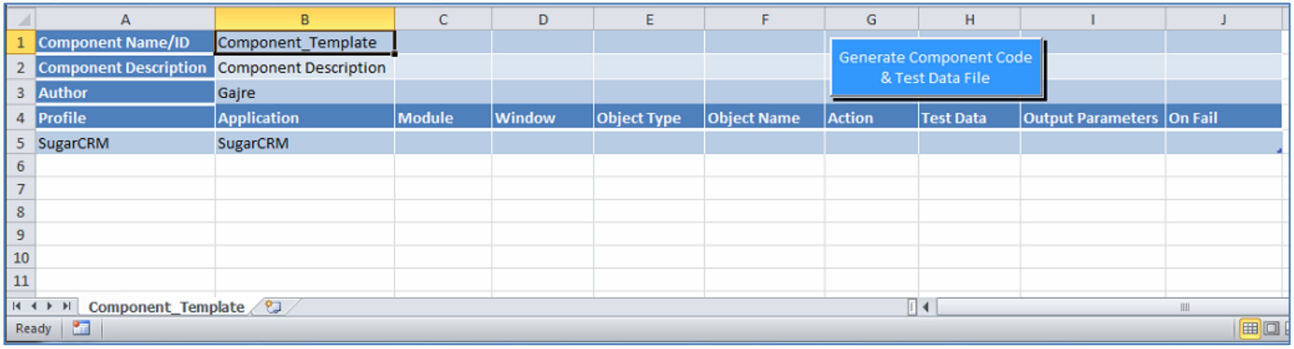
- Test Automation Framework\Components

Snapshot of the Modules under Components folder:



Accounts
Contacts
General
Leads
Component_Template.xlsm

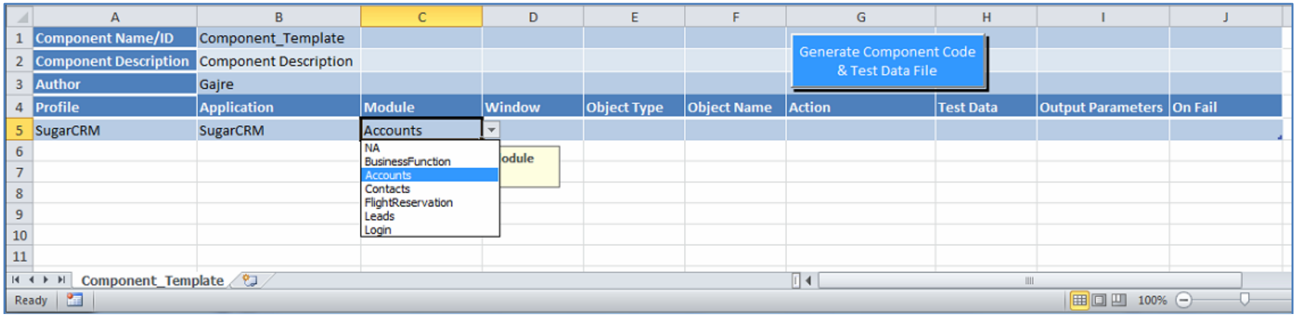
Snapshot of the Component Template:



	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM								
6										
7										
8										
9										
10										
11										

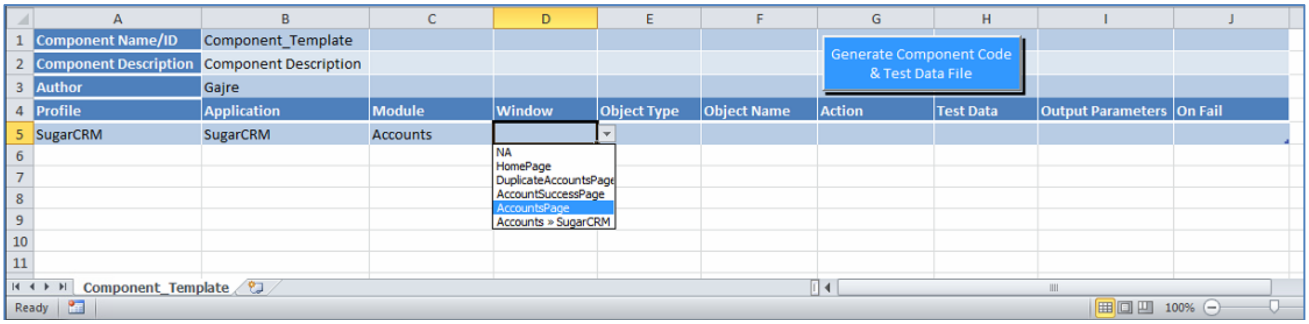
Steps to create a component:

1. Use the Component template to start designing the component.
2. Copy the "Component_Template.xlsm" to the desired module folder under Components folder.
3. Rename the file appropriately, for ex: Enter Personal Details.
4. Open the file, the following actions will automatically be performed:
    a. Rename the sheet and Component ID.
    b. Auto populates the repository names under the Module column.



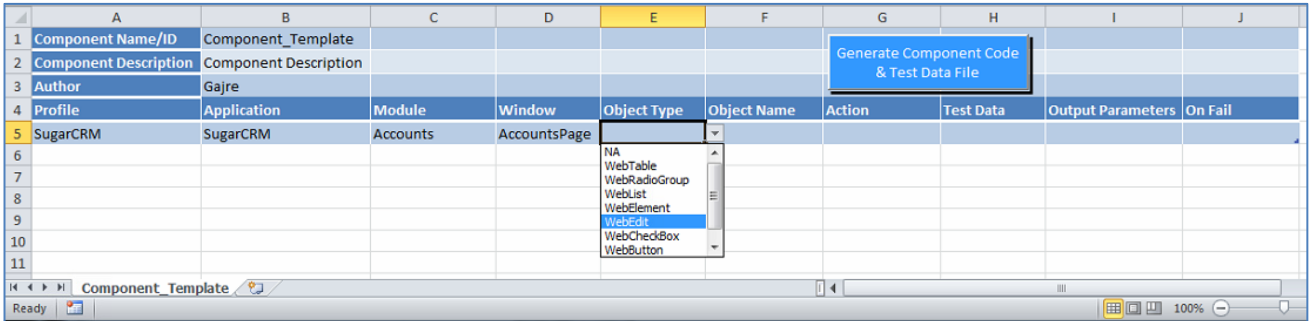
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts							
6										
7										
8										
9										
10										
11										

5. Select the repository name according to the module worked up on.
6. Based on the Module selected the Windows column will be populated with all the pages in the selected repository.



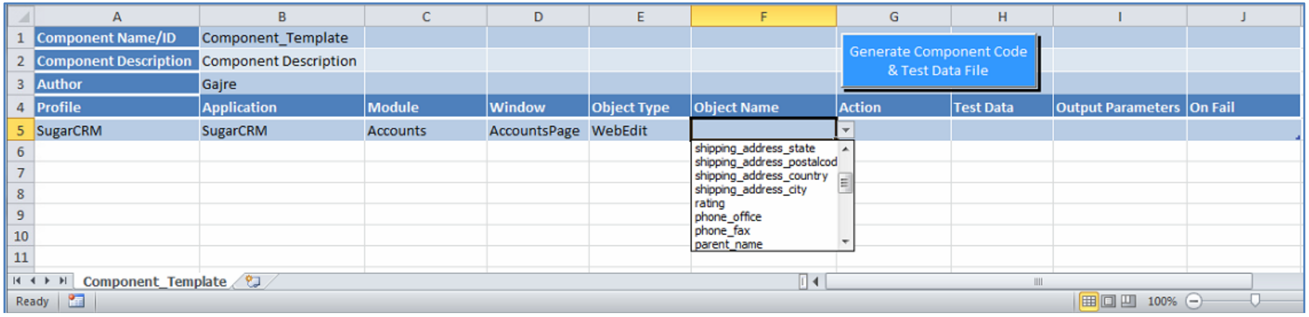
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts							

7. Based on the Window selected the Object Type column will be populated with the Object types/ Controls under the window/ page.



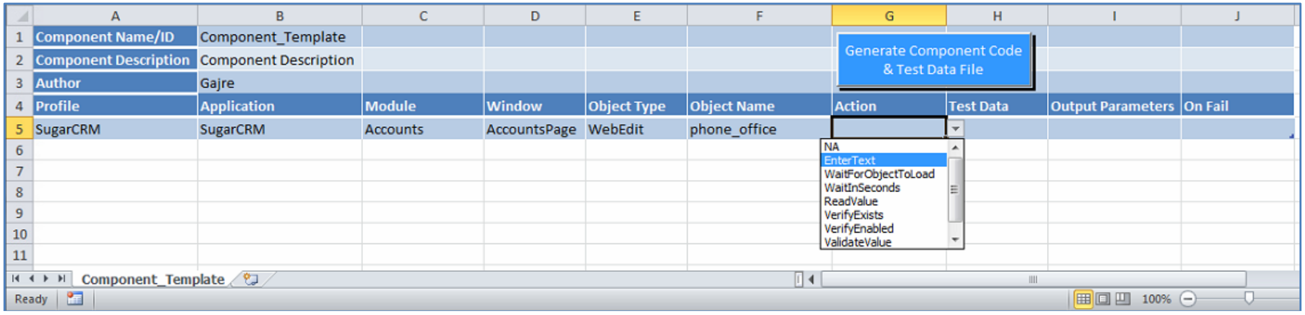
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts	AccountsPage						

8. Based on the Object type selected the Object names will be populated.



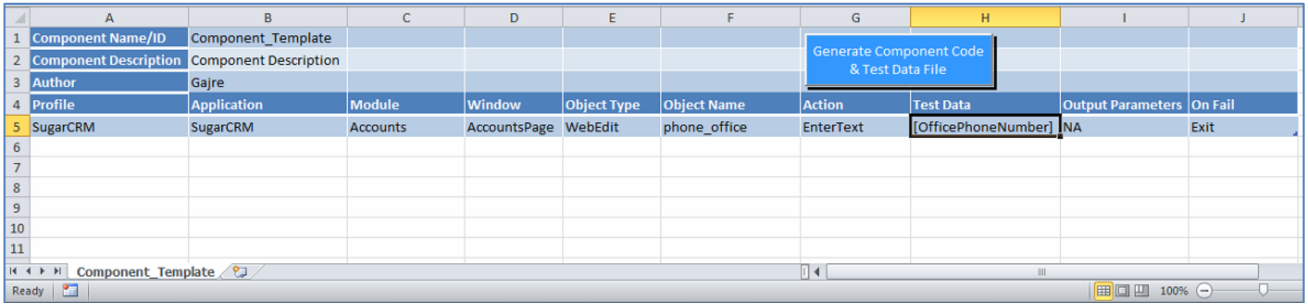
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts	AccountsPage	WebEdit					

9. After selecting the Object name, based on the Object type the Action column will be populated with respective actions.



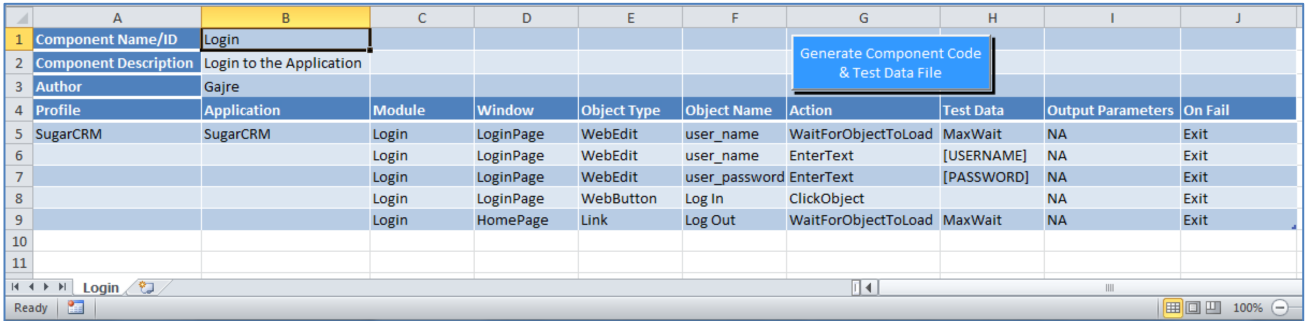
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts	AccountsPage	WebEdit	phone_office				

10. Enter the Test Data. Refer section 4.5.
    a. To pass hardcoded values just enclose the value in double quotes ex: “Search Text”.
    b. To parameterize and pass it from Test Data file, enclose the Parameter name in square braces ex: [Search Text]
    Note: Parameter name can be anything the tester wants, but it is advisable to name it w.r.t. the step or based on object name for better understanding and relating it.
11. Output Parameters value will be automatically NA, if you want to capture the output value, use the appropriate Action (ex: ReadValue) and update Output Parameters column with appropriate parameter name. ex: [Group ID]
12. On Fail, is to control the execution flow, like if you want to stop the execution if failure occur, select “Exit” and “Continue” if you want to continue the execution rather the step fail’s.



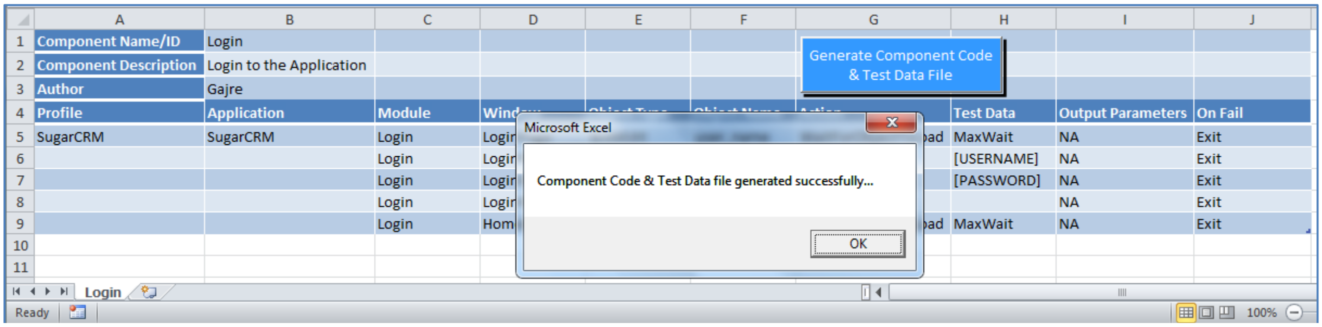
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Component_Template								
2	Component Description	Component Description								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Accounts	AccountsPage	WebEdit	phone_office	EnterText	[OfficePhoneNumber]	NA	Exit
6										
7										
8										
9										
10										
11										

13. Snapshot of a sample component created.



	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Login								
2	Component Description	Login to the Application								
3	Author	Gajre								
4	Profile	Application	Module	Window	Object Type	Object Name	Action	Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Login	LoginPage	WebEdit	user_name	WaitForObjectToLoad	MaxWait	NA	Exit
6			Login	LoginPage	WebEdit	user_name	EnterText	[USERNAME]	NA	Exit
7			Login	LoginPage	WebEdit	user_password	EnterText	[PASSWORD]	NA	Exit
8			Login	LoginPage	WebButton	Log In	ClickObject		NA	Exit
9			Login	HomePage	Link	Log Out	WaitForObjectToLoad	MaxWait	NA	Exit
10										
11										

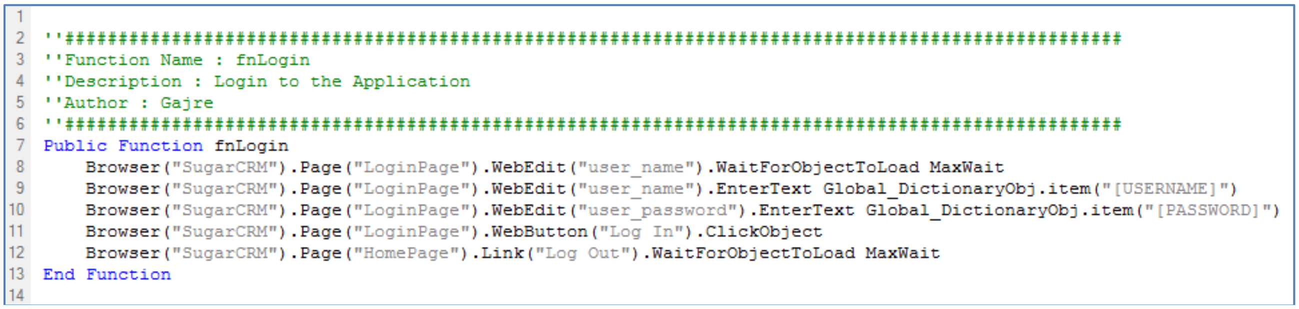
14. After the completing the Component steps design, click on the ***“Generate Component Code & Test Data File”*** button. A confirmation window is displayed on successful generation.



	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Login								
2	Component Description	Login to the Application								
3	Author	Gajre								
4	Profile	Application	Module	Wind				Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Login	Logi			ad	MaxWait	NA	Exit
6			Login	Logi				[USERNAME]	NA	Exit
7			Login	Logi				[PASSWORD]	NA	Exit
8			Login	Logi					NA	Exit
9			Login	Hom			ad	MaxWait	NA	Exit
10										
11										

Component Code & Test Data file generated successfully...

15. Automatically a method with the code for the component steps designed and a test data file with the parameter names will be created.
16. Code will be generated in the AUTLibrary file under **“*Test Automation Framework\Libraries\AUTLibraries*”** folder.

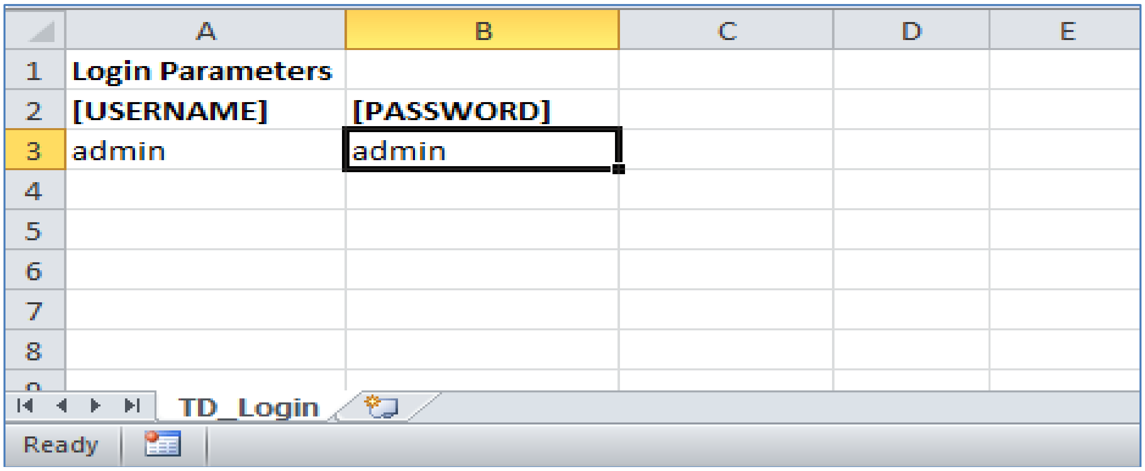


```

''###########################################################################################################
''Function Name : fnLogin
''Description : Login to the Application
''Author : Gajre
''###########################################################################################################
Public Function fnLogin
    Browser("SugarCRM").Page("LoginPage").WebEdit("user_name").WaitForObjectToLoad MaxWait
    Browser("SugarCRM").Page("LoginPage").WebEdit("user_name").EnterText Global_DictionaryObj.item("[USERNAME]")
    Browser("SugarCRM").Page("LoginPage").WebEdit("user_password").EnterText Global_DictionaryObj.item("[PASSWORD]")
    Browser("SugarCRM").Page("LoginPage").WebButton("Log In").ClickObject
    Browser("SugarCRM").Page("HomePage").Link("Log Out").WaitForObjectToLoad MaxWait
End Function

```

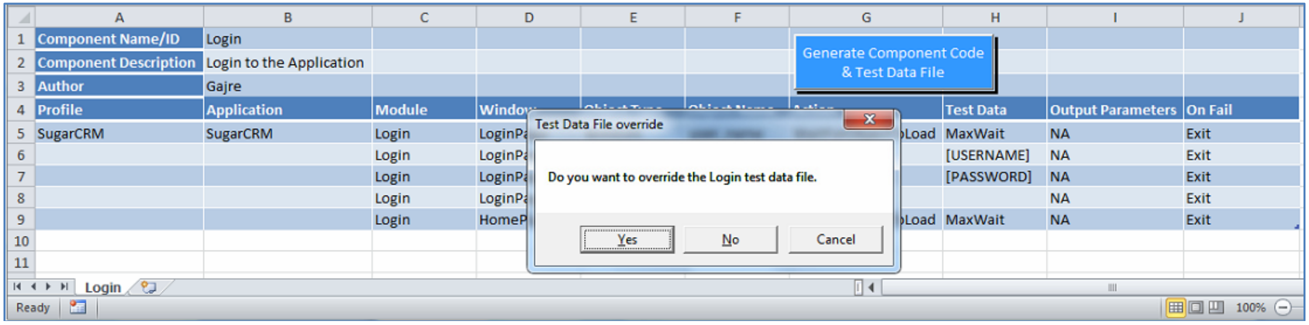
17. Test data file will be created under the “***Test Automation Framework\Test Data\<Module Name>”***



	A	B	C	D	E
1	Login Parameters				
2	[USERNAME]	[PASSWORD]			
3	admin	admin			
4					
5					
6					
7					
8					

18. User needs to update the test data file with the required test data.
19. In case the user wants to edit/ modify the component steps, he may choose do it any time and post modifications, the user needs to click ***“Generate Component Code & Test Data File”*** button to update the code and test data file.

20. User will see a new pop up message asking if the user wants to override the test data file completely.



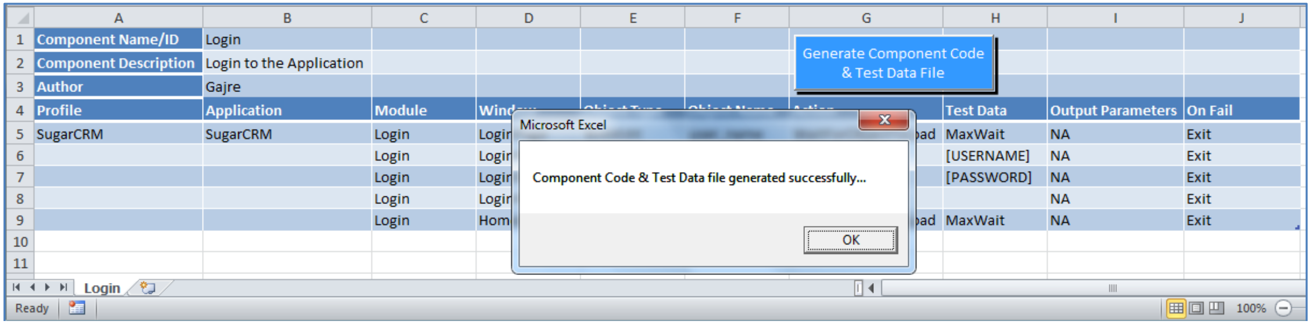
	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Login								
2	Component Description	Login to the Application								
3	Author	Gajre								
4	Profile	Application	Module	Window				Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Login	LoginPa			Load	MaxWait	NA	Exit
6			Login	LoginPa				[USERNAME]	NA	Exit
7			Login	LoginPa				[PASSWORD]	NA	Exit
8			Login	LoginPa					NA	Exit
9			Login	HomeP			Load	MaxWait	NA	Exit
10										
11										

Test Data File override

Do you want to override the Login test data file.

21. User may perform the following operations:

    a. If the user clicks “Yes”, the complete test data file will be overridden, and will see pop up with confirmation of code and test data file generated successfully.

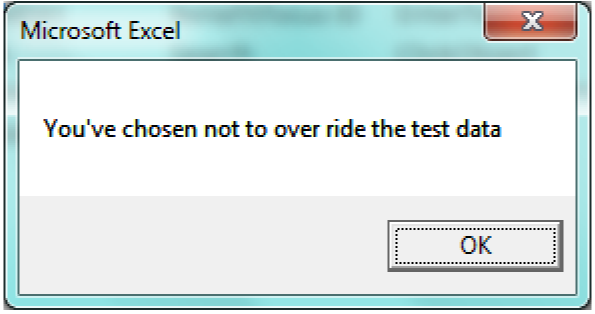


	A	B	C	D	E	F	G	H	I	J
1	Component Name/ID	Login								
2	Component Description	Login to the Application								
3	Author	Gajre								
4	Profile	Application	Module	Wind				Test Data	Output Parameters	On Fail
5	SugarCRM	SugarCRM	Login	Login			ad	MaxWait	NA	Exit
6			Login	Login				[USERNAME]	NA	Exit
7			Login	Login				[PASSWORD]	NA	Exit
8			Login	Login					NA	Exit
9			Login	Hom			ad	MaxWait	NA	Exit
10										
11										

Microsoft Excel

Component Code & Test Data file generated successfully...

    b. If the user has already designed the test data and doesn’t want the existing file to be overridden, then the user can click “No”, and the test data file will not be overridden. User will see another pop up message confirming that the user has chosen not to override the test data file. Later the user may choose to update the test data file as per the modifications or the parameters added.



Microsoft Excel

You've chosen not to over ride the test data

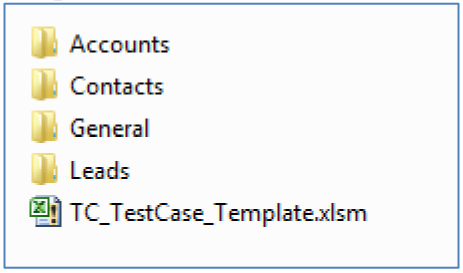
    c. The user may click “Cancel” in order to avoid the test data file from being overridden.

4.3 Test Cases

4.3.1 Folder Path in Framework

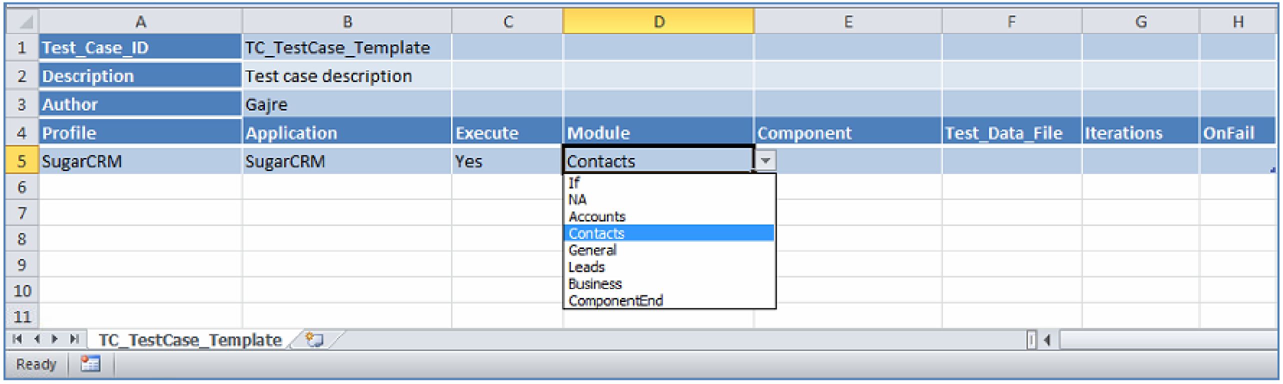
- Test Automation Framework\Test Cases

Snapshot of the Modules under Test Cases folder:



- Accounts
- Contacts
- General
- Leads
- TC_TestCase_Template.xlsm

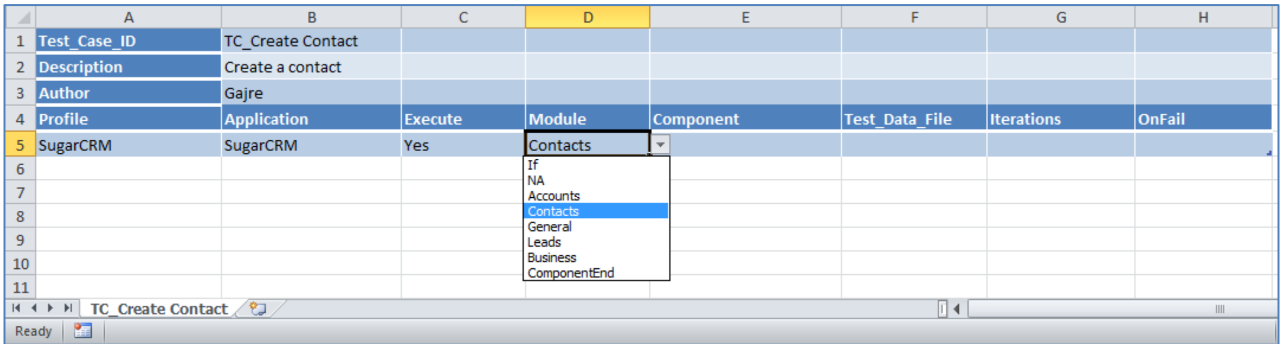
Snapshot of the Test Case Template:



	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_TestCase_Template						
2	Description	Test case description						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts				
6								
7								
8								
9								
10								
11								

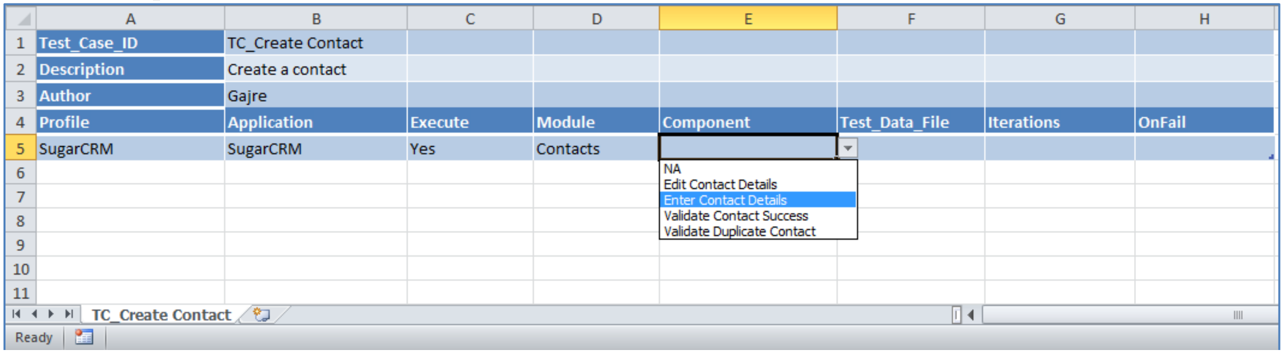
Steps to create a Test case:

1. Use the Test case template to start designing the test case.
2. Copy the "TestCase_Template.xlsm" to the desired module folder which is under Test Cases folder.
3. Rename the file appropriately, for ex: TC_EnterContactDetails.
4. Open the file, the following actions will automatically be performed:
   a. Rename the sheet and Test_Case_ID.
   b. Auto populates the Module column with the module folder names under Component folder.



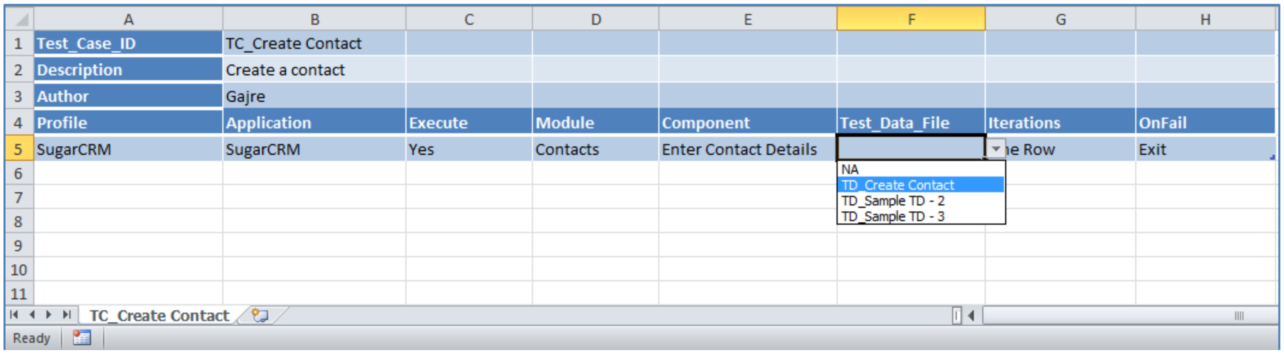
	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts				
6								
7								
8								
9								
10								
11								

5. Based on the Module selected the Component column will be populated with all the components under the Module folder.



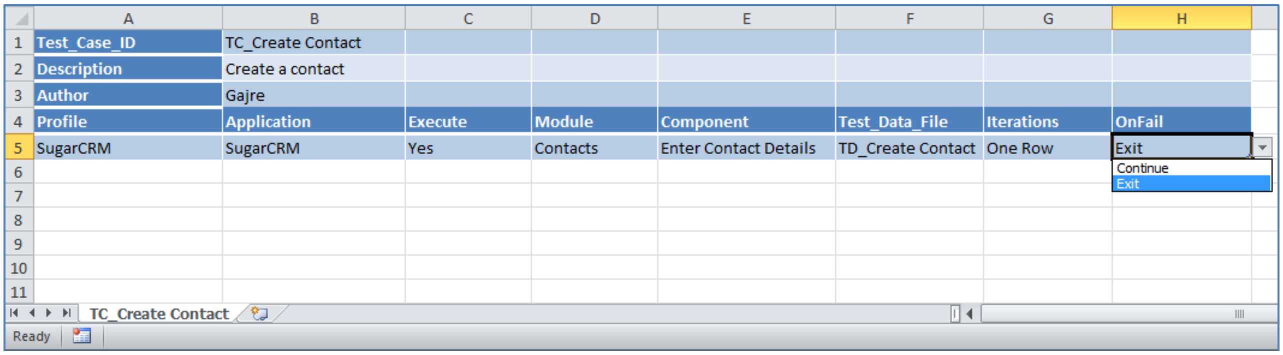
	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts				
6								
7								
8								
9								
10								
11								

6. After the Component is selected, based on the Module selected the Test_Data_File column will be populated with all the Test Data files under the respective Module folder under Test Data folder.



	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts	Enter Contact Details		e Row	Exit
6								
7								
8								
9								
10								
11								

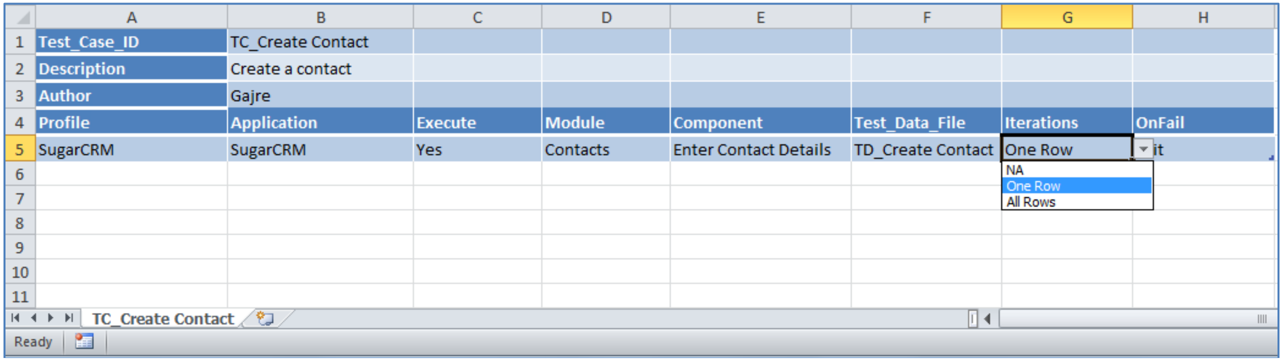
7. Select the appropriate Test_Data_File or select “NA” if test data doesn’t apply or is not required for the respective component.
8. OnFail, is to control the execution flow, like if you want to stop the execution if failure occurs, select “Exit” and “Continue” if you want to continue the execution even if the Component fail’s.



	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts	Enter Contact Details	TD_Create Contact	One Row	Exit

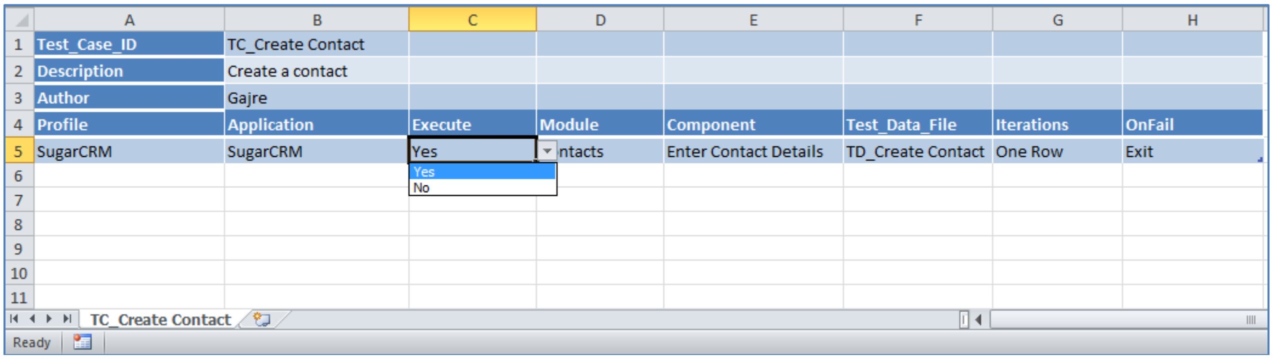
9. In the Iterations column, select the no. of iterations you want the component to execute for.
    a. Select “One Row” if you desire to execute the component for one set of test data in the test data file.
    b. Select “All Rows” if you desire to execute the component for all sets of test data in the test data file.
    c. Select “NA” if the test data file is marked “NA”.

    Note: If you want you may use the same test data file for other test case steps as well. To do that, just select the same data file from the drop down and select the iterations accordingly.



	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	Contacts	Enter Contact Details	TD_Create Contact	One Row	it

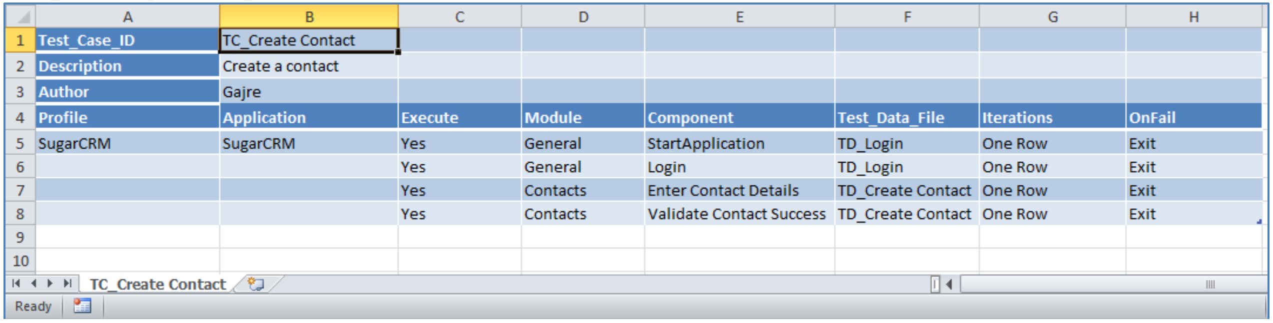
10. In the Execute column:
    a. Select “Yes” against the components if you want them to be executed as part of the test case.
    b. Select “No” if you don’t want the component to be executed as part of the test case.



	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes Yes No	ntacts	Enter Contact Details	TD_Create Contact	One Row	Exit
6								
7								
8								
9								
10								
11								

11. Write the Description and Author name in the respective fields.

Snapshot of a completed Test Case:



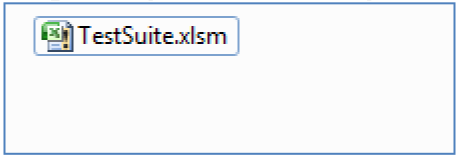
	A	B	C	D	E	F	G	H
1	Test_Case_ID	TC_Create Contact						
2	Description	Create a contact						
3	Author	Gajre						
4	Profile	Application	Execute	Module	Component	Test_Data_File	Iterations	OnFail
5	SugarCRM	SugarCRM	Yes	General	StartApplication	TD_Login	One Row	Exit
6			Yes	General	Login	TD_Login	One Row	Exit
7			Yes	Contacts	Enter Contact Details	TD_Create Contact	One Row	Exit
8			Yes	Contacts	Validate Contact Success	TD_Create Contact	One Row	Exit
9								
10								

4.4 Test Suite

4.4.1 Folder Path in Framework

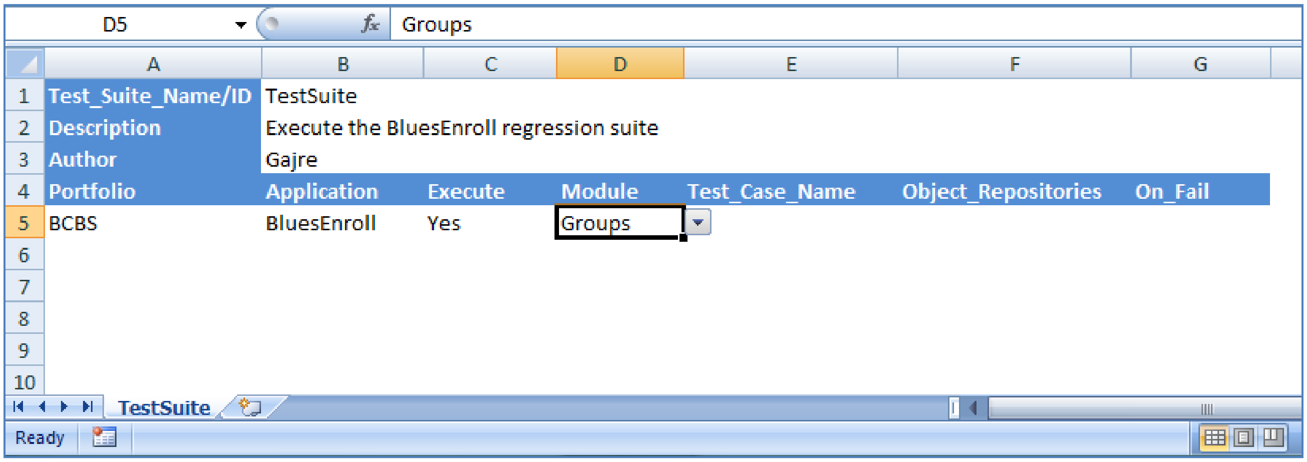
- Test Automation Framework\Test Suite

Snapshot of the Test Suites folder with "TestSuite.xlsm" file:



TestSuite.xlsm

Snapshot of the Test Suite Template:

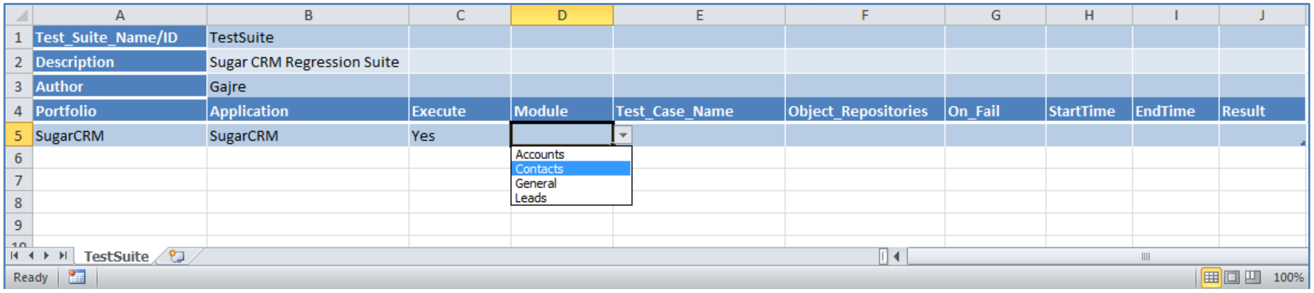


D5 Groups

	A	B	C	D	E	F	G
1	Test_Suite_Name/ID	TestSuite					
2	Description	Execute the BluesEnroll regression suite					
3	Author	Gajre					
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail
5	BCBS	BluesEnroll	Yes	Groups			

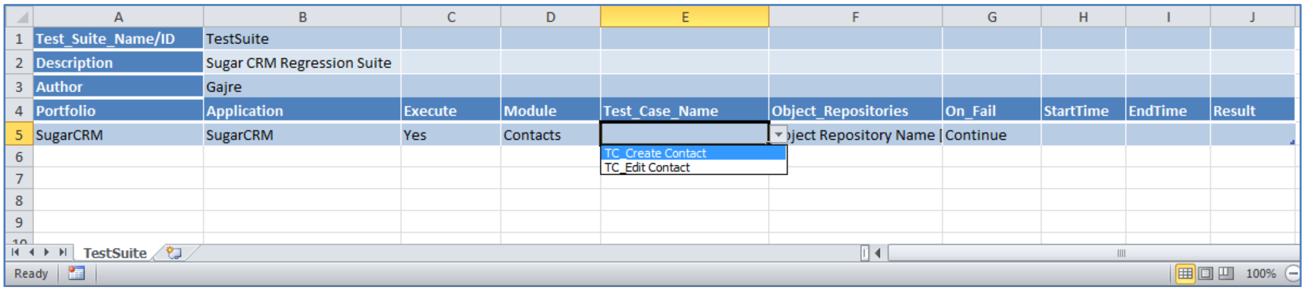
Steps to create a Test case:

1. Use the Test Suite.xlsm file to start designing the test suite.
2. Open the file, the following actions will automatically be performed:
    a. Rename the sheet and Test_Suite_Name/ID.
    b. Auto populates the Module column with the module folder names under Test Cases folder.
    Note: The name of the test suite file should be always ***"TestSuite.xlsm"***



	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes							

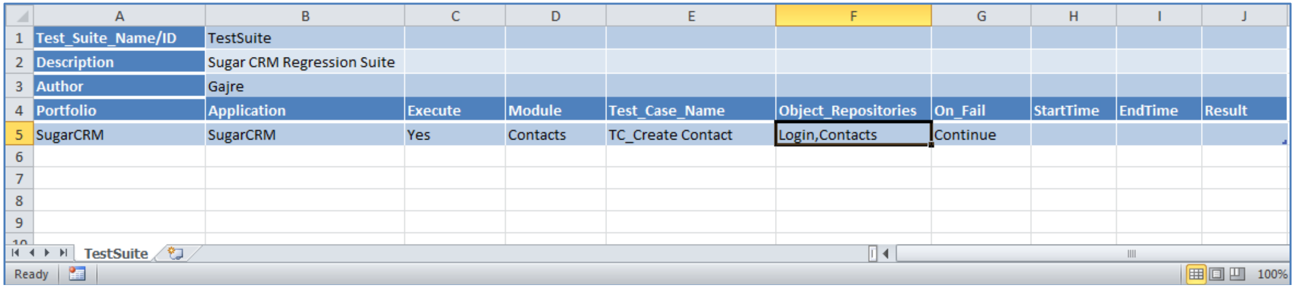
3. Based on the Module selected the Test_Case_Name column will be populated with all the Test cases under the Module folder.



	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes	Contacts		ject Repository Name	Continue			

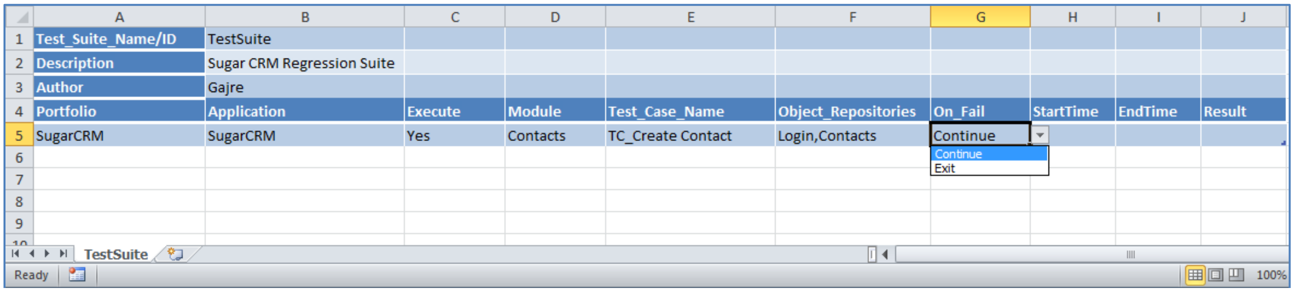
4. Select the Test Case in the Test_Case_Name column.
5. In the Object_Repository column, write the Object repository name required for the Test case. If multiple Object repositories are required separate them with a comma

“,”. To know what Object repositories are to be mentioned here, you may quickly refer the components (module column info) used to design the test case.



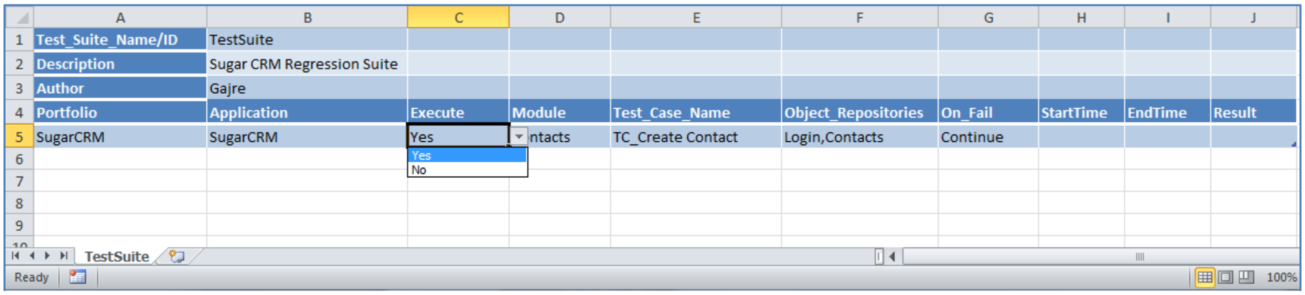
	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes	Contacts	TC_Create Contact	Login,Contacts	Continue			
6										
7										
8										
9										

6. On_Fail is to control the execution flow, like if you want to stop the execution if failure occurs, select “Exit” and “Continue” if you want to continue the execution even if the Test case fail’s.



	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes	Contacts	TC_Create Contact	Login,Contacts	Continue			
6										
7										
8										
9										

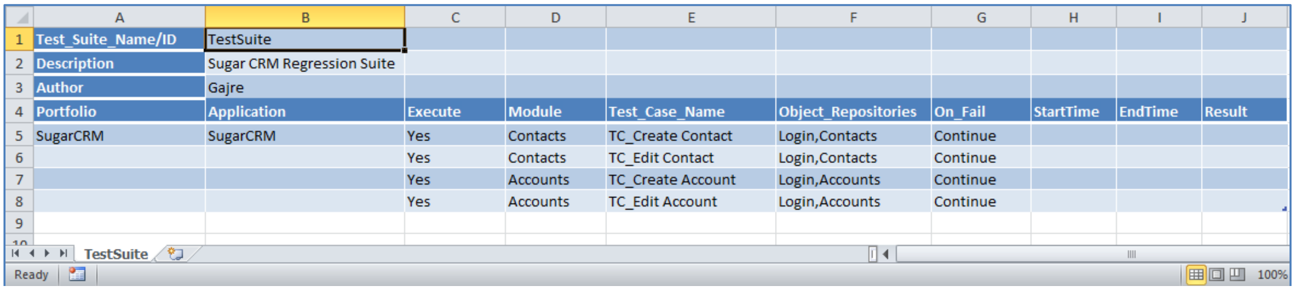
7. In the Execute column:
    a. Select “Yes” against the test cases if you want them to be executed as part of the test suite.
    b. Select “No” if you don’t want the test cases to be executed as part of the test suite.



	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes	ntacts	TC_Create Contact	Login,Contacts	Continue			
6										
7										
8										
9										

8. Write the Description and Author name in the respective fields.

Snapshot of a Test Suite:



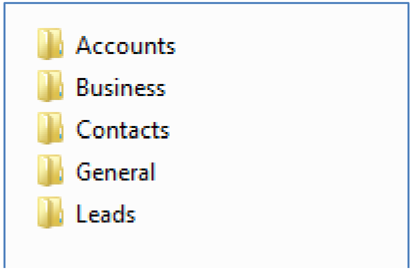
	A	B	C	D	E	F	G	H	I	J
1	Test_Suite_Name/ID	TestSuite								
2	Description	Sugar CRM Regression Suite								
3	Author	Gajre								
4	Portfolio	Application	Execute	Module	Test_Case_Name	Object_Repositories	On_Fail	StartTime	EndTime	Result
5	SugarCRM	SugarCRM	Yes	Contacts	TC_Create Contact	Login,Contacts	Continue			
6			Yes	Contacts	TC_Edit Contact	Login,Contacts	Continue			
7			Yes	Accounts	TC_Create Account	Login,Accounts	Continue			
8			Yes	Accounts	TC_Edit Account	Login,Accounts	Continue			
9										

4.5 Test Data

4.5.1 Folder Path in Framework

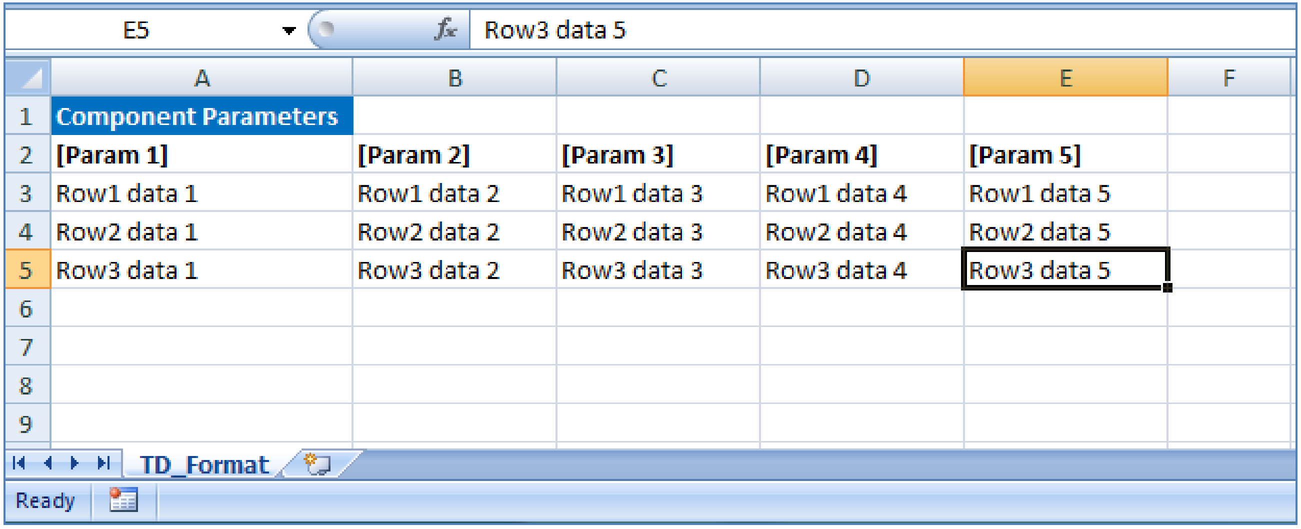
- Test Automation Framework\Test Data

Snapshot of the Modules under Test Suites folder:



- Accounts
- Business
- Contacts
- General
- Leads

Snapshot of the Test Data Templates/ formats:
Normal test data format:

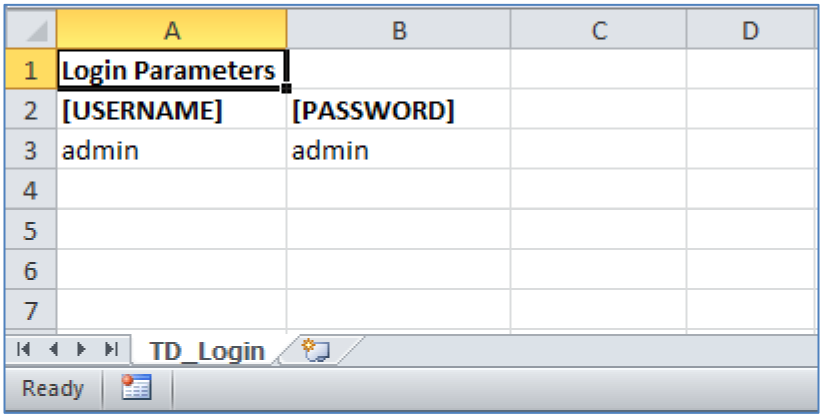


E5 fx Row3 data 5

	A	B	C	D	E	F
1	Component Parameters					
2	[Param 1]	[Param 2]	[Param 3]	[Param 4]	[Param 5]	
3	Row1 data 1	Row1 data 2	Row1 data 3	Row1 data 4	Row1 data 5	
4	Row2 data 1	Row2 data 2	Row2 data 3	Row2 data 4	Row2 data 5	
5	Row3 data 1	Row3 data 2	Row3 data 3	Row3 data 4	Row3 data 5	
6						
7						
8						
9						

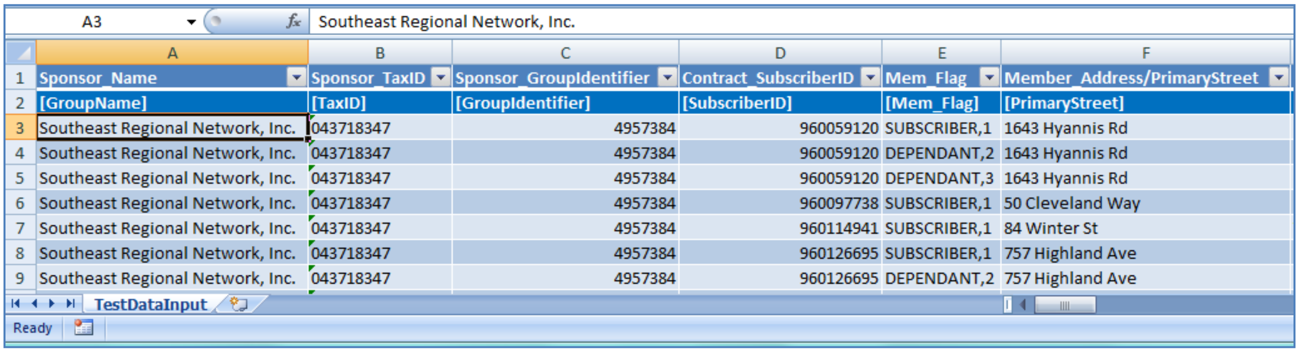
Steps to create a Test data:

1. The above formats are designed to accommodate test data from XML files and the local test data designed by user.
2. In the above format:
   a. 1st row should be description of the test data or the XML node information.
   b. 2nd row if the parameter names in both the cases.
   c. From 3rd row is the test data.
3. **Normal test data format** will be created, when we click on the ***"Generate Component Code & Test Data File"*** button after completing the Component steps design. It directly takes the Parameters used in the components and creates the test data file.
4. User needs to enter the test data from the 3rd row, in the normal format.



	A	B	C	D
1	Login Parameters			
2	[USERNAME]	[PASSWORD]		
3	admin	admin		
4				
5				
6				
7				

5. To generate the test data from XML input file, user needs to place the latest XML file in the following path *"Test Automation Framework\Test Data\XML"* and double click the "*XMLDataToXLData.exe"* in the "*Test Automation Framework*" folder. This will generate the test data in XML test data format in the following path: "*Test Automation Framework\Test Data\Business"*.



A3 fx Southeast Regional Network, Inc.

	A	B	C	D	E	F
1	Sponsor_Name	Sponsor_TaxID	Sponsor_GroupIdentifier	Contract_SubscriberID	Mem_Flag	Member_Address/PrimaryStreet
2	[GroupName]	[TaxID]	[GroupIdentifier]	[SubscriberID]	[Mem_Flag]	[PrimaryStreet]
3	Southeast Regional Network, Inc.	043718347	4957384	960059120	SUBSCRIBER,1	1643 Hyannis Rd
4	Southeast Regional Network, Inc.	043718347	4957384	960059120	DEPENDANT,2	1643 Hyannis Rd
5	Southeast Regional Network, Inc.	043718347	4957384	960059120	DEPENDANT,3	1643 Hyannis Rd
6	Southeast Regional Network, Inc.	043718347	4957384	960097738	SUBSCRIBER,1	50 Cleveland Way
7	Southeast Regional Network, Inc.	043718347	4957384	960114941	SUBSCRIBER,1	84 Winter St
8	Southeast Regional Network, Inc.	043718347	4957384	960126695	SUBSCRIBER,1	757 Highland Ave
9	Southeast Regional Network, Inc.	043718347	4957384	960126695	DEPENDANT,2	757 Highland Ave

6. The driver script doesn't need any special instructions/ inputs to understand the formats. It treats both similar and reads the test data dynamically.

4.6 Triggering the executions

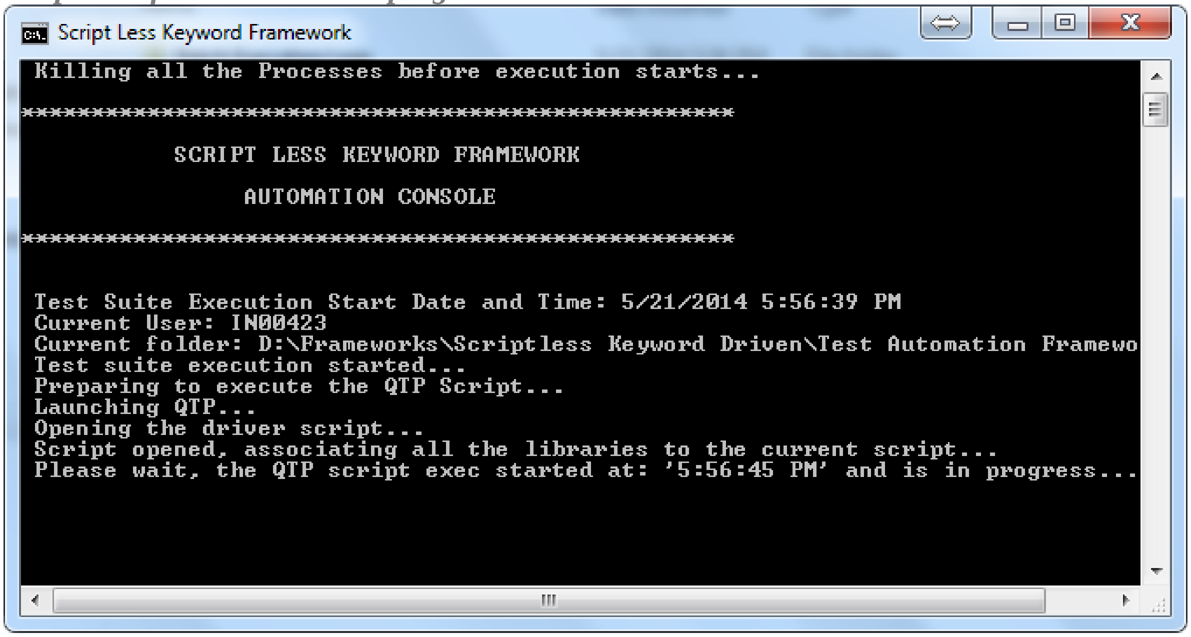
4.6.1 Batch file path in Framework

- Test Automation Framework\RunBatch.bat

Steps to create a Test data:

1. Once the *"TestSuite.xlsm"* is ready with the test cases information for execution for ex: designed as part of regression suite.
2. Double click the *"RunBatch.bat"* file under *"Test Automation Framework"* folder.
3. The execution will be triggered and user will see a command prompt with the progress of execution.

Snapshot of the execution in progress:



```
 Killing all the Processes before execution starts...

***************************************************

           SCRIPT LESS KEYWORD FRAMEWORK

                AUTOMATION CONSOLE

***************************************************


 Test Suite Execution Start Date and Time: 5/21/2014 5:56:39 PM
 Current User: IN00423
 Current folder: D:\Frameworks\Scriptless Keyword Driven\Test Automation Framewo
 Test suite execution started...
 Preparing to execute the QTP Script...
 Launching QTP...
 Opening the driver script...
 Script opened, associating all the libraries to the current script...
 Please wait, the QTP script exec started at: '5:56:45 PM' and is in progress...
```

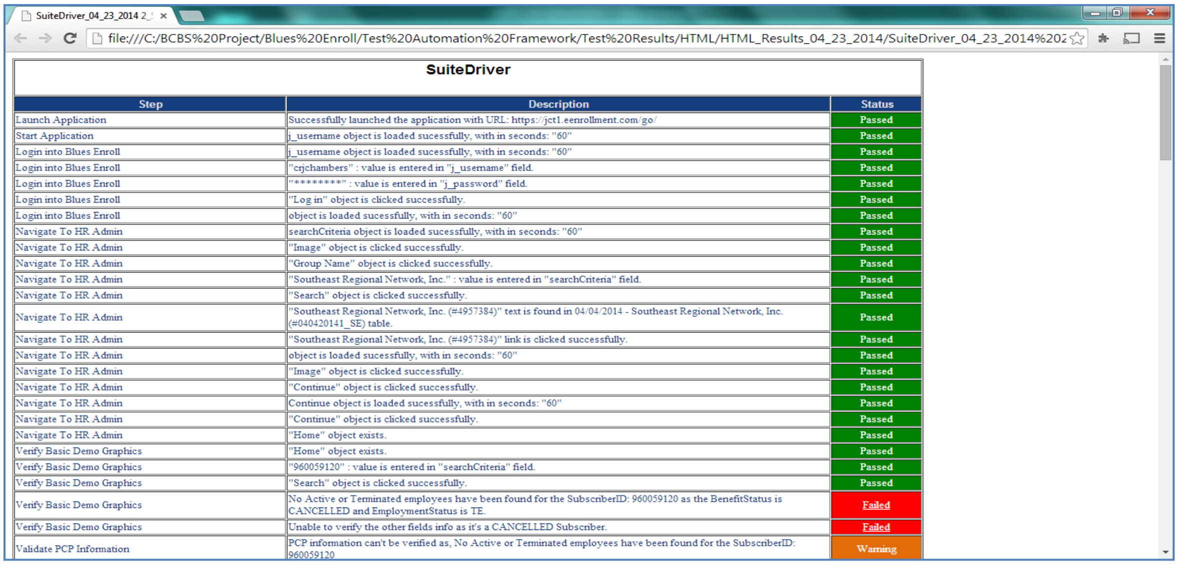
4. In the *"TestSuite.xlsm"* all the test cases information will be read one by one, followed by each component information in the particular test case in a sequence.
5. Post execution completion the user will see the notification in the command prompt and the test results will be saved in the following path: "*Test Automation Framework\Test Results\HTML\<HTML_Results_Date Stamp>"*.

4.7 Test Results

4.7.1 Folder path in Framework

- Test Automation Framework\Test Results\HTML\<HTML_Results_Date Stamp>

Snapshot of the Test Results:

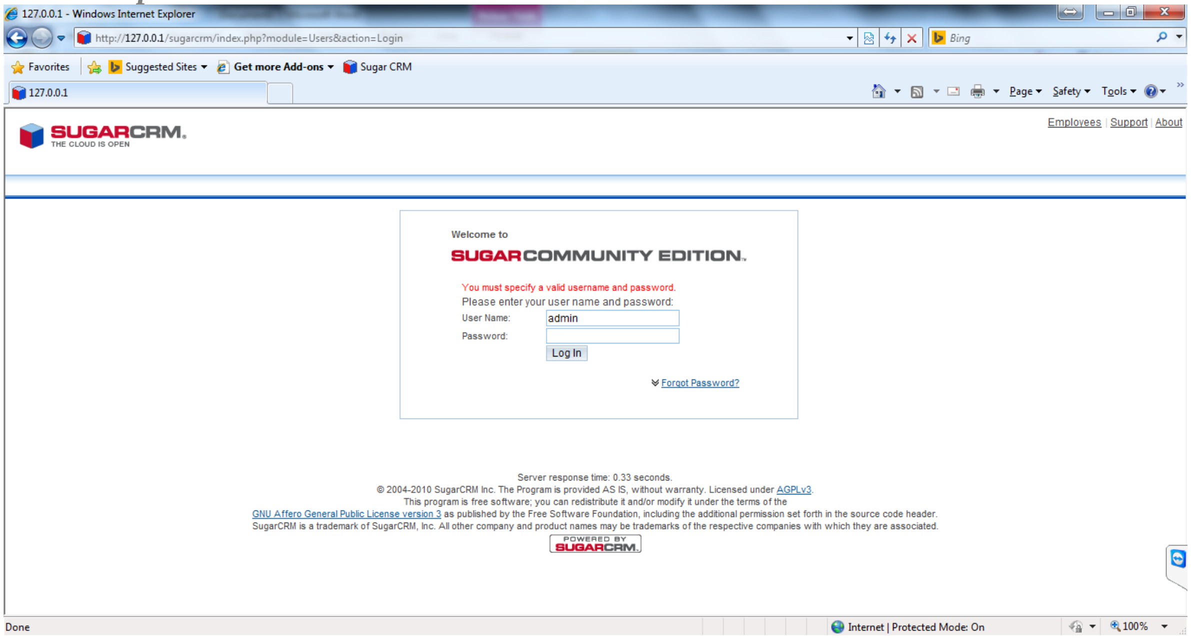

SuiteDriver_04_23_2014 2_
file:///C:/BCBS%20Project/Blues%20Enroll/Test%20Automation%20Framework/Test%20Results/HTML/HTML_Results_04_23_2014/SuiteDriver_04_23_2014%202

SuiteDriver

Step	Description	Status
Launch Application	Successfully launched the application with URL: https://jct1.eenrollment.com/go/	Passed
Start Application	j_username object is loaded sucessfully, with in seconds: "60"	Passed
Login into Blues Enroll	j_username object is loaded sucessfully, with in seconds: "60"	Passed
Login into Blues Enroll	"crjchambers" : value is entered in "j_username" field.	Passed
Login into Blues Enroll	"********" : value is entered in "j_password" field.	Passed
Login into Blues Enroll	"Log in" object is clicked successfully.	Passed
Login into Blues Enroll	object is loaded sucessfully, with in seconds: "60"	Passed
Navigate To HR Admin	searchCriteria object is loaded sucessfully, with in seconds: "60"	Passed
Navigate To HR Admin	"Image" object is clicked successfully.	Passed
Navigate To HR Admin	"Group Name" object is clicked successfully.	Passed
Navigate To HR Admin	"Southeast Regional Network, Inc." : value is entered in "searchCriteria" field.	Passed
Navigate To HR Admin	"Search" object is clicked successfully.	Passed
Navigate To HR Admin	"Southeast Regional Network, Inc. (#4957384)" text is found in 04/04/2014 - Southeast Regional Network, Inc. (#040420141_SE) table.	Passed
Navigate To HR Admin	"Southeast Regional Network, Inc. (#4957384)" link is clicked successfully.	Passed
Navigate To HR Admin	object is loaded sucessfully, with in seconds: "60"	Passed
Navigate To HR Admin	"Image" object is clicked successfully.	Passed
Navigate To HR Admin	"Continue" object is clicked successfully.	Passed
Navigate To HR Admin	Continue object is loaded sucessfully, with in seconds: "60"	Passed
Navigate To HR Admin	"Continue" object is clicked successfully.	Passed
Navigate To HR Admin	"Home" object exists.	Passed
Verify Basic Demo Graphics	"Home" object exists.	Passed
Verify Basic Demo Graphics	"960059120" : value is entered in "searchCriteria" field.	Passed
Verify Basic Demo Graphics	"Search" object is clicked successfully.	Passed
Verify Basic Demo Graphics	No Active or Terminated employees have been found for the SubscriberID: 960059120 as the BenefitStatus is CANCELLED and EmploymentStatus is TE.	Failed
Verify Basic Demo Graphics	Unable to verify the other fields info as it's a CANCELLED Subscriber.	Failed
Validate PCP Information	PCP information can't be verified as, No Active or Terminated employees have been found for the SubscriberID: 960059120	Warning

1. If any errors occur during the execution the screenshots will be taken and stored in the following path: "*Test Automation Framework\Test Results\HTML\<HTML_Results_Date Stamp>\ ErrorSnapShot*".
2. The failed steps in the HTML result file will be marked red and will be a "Failed" link with the reference to the respective error screenshot.
3. Click the respective step "Failed" link to see the error screenshot.

Error Snapshot:



Welcome to
SUGARCOMMUNITY EDITION.

You must specify a valid username and password.
Please enter your user name and password:
User Name: admin
Password:

Server response time: 0.33 seconds.
© 2004-2010 SugarCRM Inc. The Program is provided AS IS, without warranty. Licensed under AGPLv3.
This program is free software; you can redistribute it and/or modify it under the terms of the
GNU Affero General Public License version 3 as published by the Free Software Foundation, including the additional permission set forth in the source code header.
SugarCRM is a trademark of SugarCRM, Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

5. *Glossary*

Item	Definition