



AFSG

(Assurant Framework Step Generator)

User Manual

Prepared for
AEB, Kansas

Table of Contents

- Section 1: About This Document 4
- Section2: Step Generator Artifacts 5
 - Overview5
 - Types of Artifacts.....5
 - Folder Structure.....7
- Section3. Software..... 8
 - Prerequisites.....8
 - Browser Settings8
- Section4.Using Step Generator... 11
 - 4.1Prerequisites 11
 - 4.2 Login to Step Generator 11
 - 4.3 Application Maintenance 12
 - 4.3.1 Select Application 12
 - 4.3.2 Add Application 12
 - 4.3.3 Edit Application 13
 - 4.3.4 Delete Application..... 13
 - 4.4 Object Repository Management..... 15
 - 4.4.1 Adding Objects in Object Repository 15
 - 4.4.2 Editing Objects in Object Repository 17
 - 4.4.3 Deleting Objects in Object Repository 19
 - 4.5 Action Management 21
 - 4.5.1 Adding Actions in Step Generator 21
 - 4.5.2 Editing Actions in Step Generator 23
 - 4.5.3 Deleting Actions in Step Generator 25
 - 4.6 Function Library Management 27
 - 4.6.1 Adding Function Library in Step Generator 27
 - 4.6.2 Editing Function Library path in Step Generator 29
 - 4.6.3 Deleting Function Library in Step Generator 31
 - 4.7 Selecting existing Test Case (.xml) 33
 - 4.8 Starting with new Test Case 35
 - 4.9 Add Step Tab 36
 - 4.9.1 Writing Test Case 36
 - 4.9.1.1 Using the GUI 36
 - 4.9.1.2 Using the Web-Table 37
 - 4.9.1.3 Re-Usable Scripts 39
 - 4.9.1.4 User Defined Functions 39
 - 4.9.2 Deleting Test Steps 40
 - 4.9.2.1 Deleting Single Test Step 40
 - 4.9.2.2 Deleting multiple steps from Test Case 40
 - 4.9.2.3 Deleting all steps from Test Case 41
 - 4.9.3 Saving Test Case 42
 - 4.9.3.1 Saving to temp File 42
 - 4.9.3.2 Saving to custom folder 42
 - 4.9.4 Exporting Test Case (In Excel) 43
 - 4.10 Execute Script Tab 44
 - 4.10.1 Understanding Execute Script Tab 44

4.10.2 Executing Script..... 45

4.10.2.1 Selecting the Excel file..... 45

4.10.2.2 Executing after writing script in Utility..... 47

4.10.3 Analyzing results..... 48

Section 1: About This Document

The Framework Step Generator User Manual is designed with a purview of answering the most frequently asked question – “*How does Step Generator –Work?*”

The document highlights on step generator utility usage and its advantages. The document also speaks about to usage of the Step Generator to create; update and deletion of test steps in automation script along with object management and script execution.

Section2: Step Generator Artifacts

Overview

The framework step generator uses different types of test artifacts to create; update and deletion of objects and test steps for new test scripts. They are Admin login, Step Creation, Add Object and Delete/Update Object. All these artifacts are **stored** in i-ASTG folder.

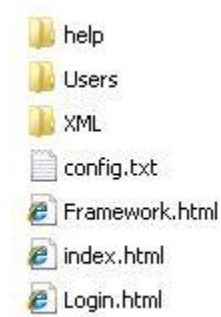
Types of Artifacts

Test Artifacts	Description	Location	Can be modified?
Login	The main page where user uses his/her LAN ID and password to login to the application	<i>T:\Test Services\Offshore Folder\Final_VI\HTML PRoj/Login.html</i>	Yes. Only user having access to QC Project: AEB_AUTOMATION and Domain: Web_Automation can login.
Index Page <ul style="list-style-type: none">• Select Application• Add Application• Update Application• Delete Application	This page contains “Select Applications, object Repository and Test Script Path browse button” i.e. Available Applications for automation along with Application Object Repository path and Test Script Path (XML).	<i>E.g.:</i> <i>Application Name: SAS</i> <i>Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\XML\Application OR\SAS\SAS.xml</i> <i>Test Script Path: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\lr1385\temp\temp.xml</i>	Yes. Select Application and Test Script path can be modified.
Framework Page <ul style="list-style-type: none">• Add Step• Object Repository• Action Management• Function Library Management• Execute Script	Add Step tab contains Test Case details i.e. Test Case Id, objects details and its respective action list and data parameter.		Yes. Add Step for automation can be added and it can be modified.
	Object Repository tab contains the list of the properties to be added in to the object.		Yes. Adding the objects with different properties can be achieved.
	Action Management tab is to manage actions i.e. new keywords to add or to change the name of the existing actions.		Yes. Adding the actions can be achieved.
	Function Library Management tab is to manage the paths of function library.		Yes. Adding the function libraries along with paths can be achieved.
	Execute Script tab is to execute scripts.		Yes. User can execute script from this tab by passing the Test Case ID.
Config.txt	This Properties file can be accessed using notepad and configure the settings for the paths of the different files.	<i>//config.txt</i>	Yes. We need to make sure that all the settings in this file are valid before executing a test case. Values can be changed by opening file from the login Page.

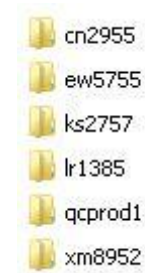
Test Artifacts	Description	Location	Can be modified?
Temp.xml	This XML contains the temporary test case of the test cases to be executed.	\\Users\ <User Name>\ Temp\ temp.xml e.g. For <i>Ir1385</i> application: \\Users\Ir1385\Temp\temp.xml	No.
Actions.XML	This XML contains the set of actions.	XML\Actions.xml	Yes. From application itself.
ApplicationsList.XML	This XML contains the set of Applications.	XML\ ApplicationsList.xml	Yes. From application itself.
Class.XML	This XML contains the set of Classes and classID.	XML\ Class.xml	Yes. From application itself.
FunctionLibraries.XML	This XML contains the set of Function Libraries and path of Function Libraries.	XML\ FunctionLibraries.xml	Yes. From application itself.

Folder Structure

Main Folder \\StepGenerator



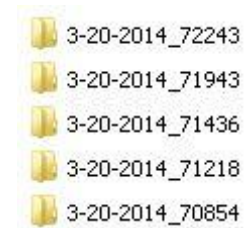
User Folder \\ StepGenerator \Users



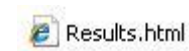
User Folder \\ StepGenerator \Users\lr1385



Results Folder \\ StepGenerator \Users\lr1385\Results



Result File \\ StepGenerator \Users\lr1385\Results\



Temp Folder \\ StepGenerator \Users\lr1385\temp\



Folder	Description
Users	<ul style="list-style-type: none">Individual user level folder.Used to manage files for user level.
Results	<ul style="list-style-type: none">Used to store execution results. Date and Time stamp is used to differentiate.Ex: 3-20-2014_72243
Temp	<ul style="list-style-type: none">Temporary folder to save the current xml and xls.Contains temp.xml and teml.xls

Section3. Software

Prerequisites

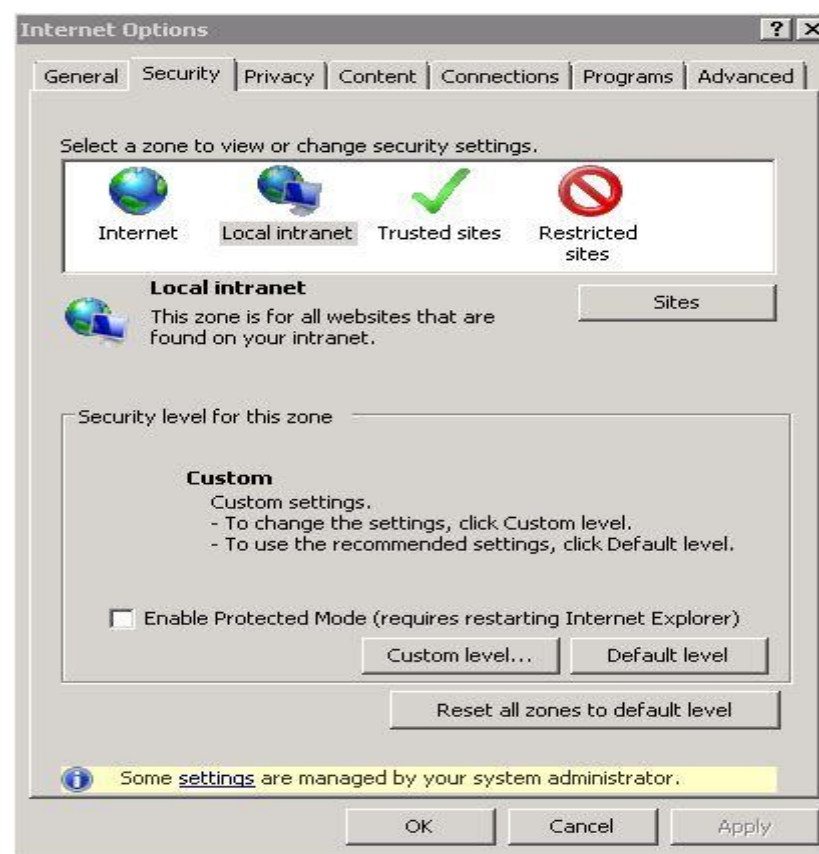
- 3.1.1 Windows XP/ windows 7
- 3.1.2 HP QTP
- 3.1.3 HP QC Credentials
- 3.1.4 Internet Explorer Version 8
- 3.1.5 Access to "\\T Drive"

Browser Settings

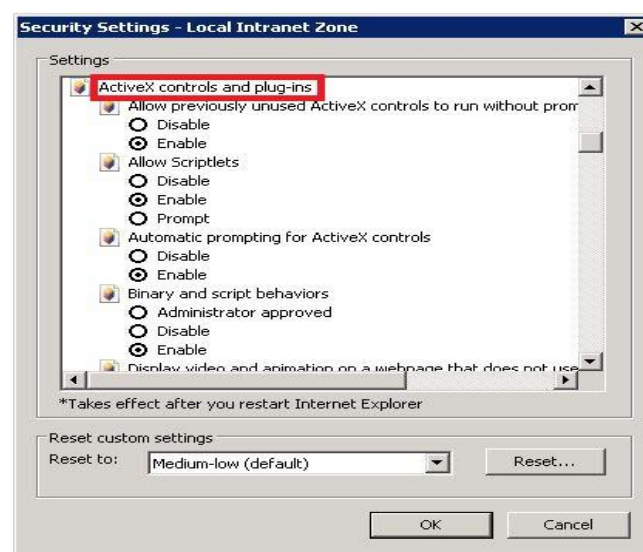
Note: Supported Browsers: IE8. Following settings are related to IE8

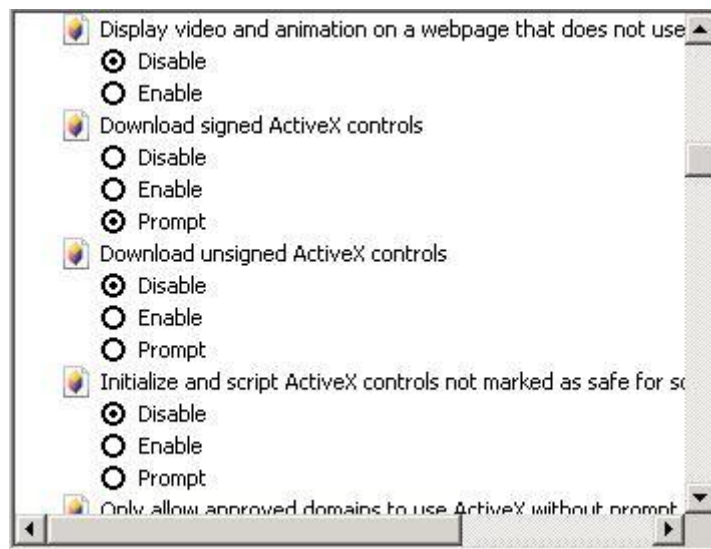
Step 1: Go to Tools ->Internet Options.

Step2: Click on the **Local internet** symbol and click on Custom Level button

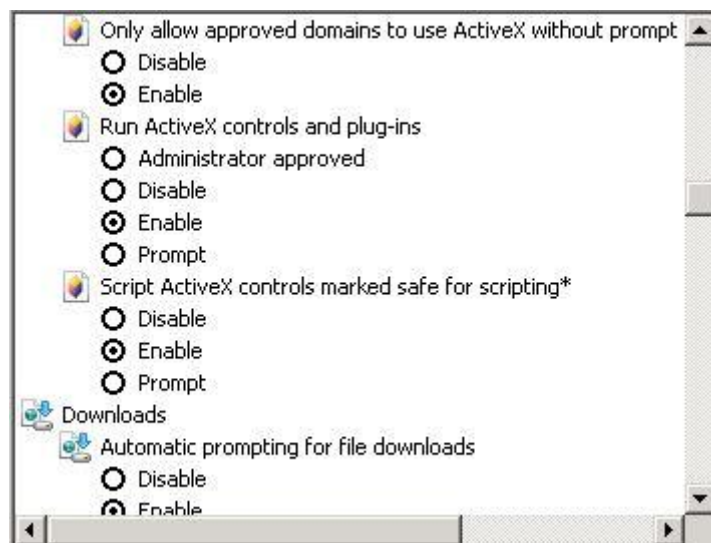


Step 3: In the **ActiveX controls and plug-ins** options mark everything "enabled".

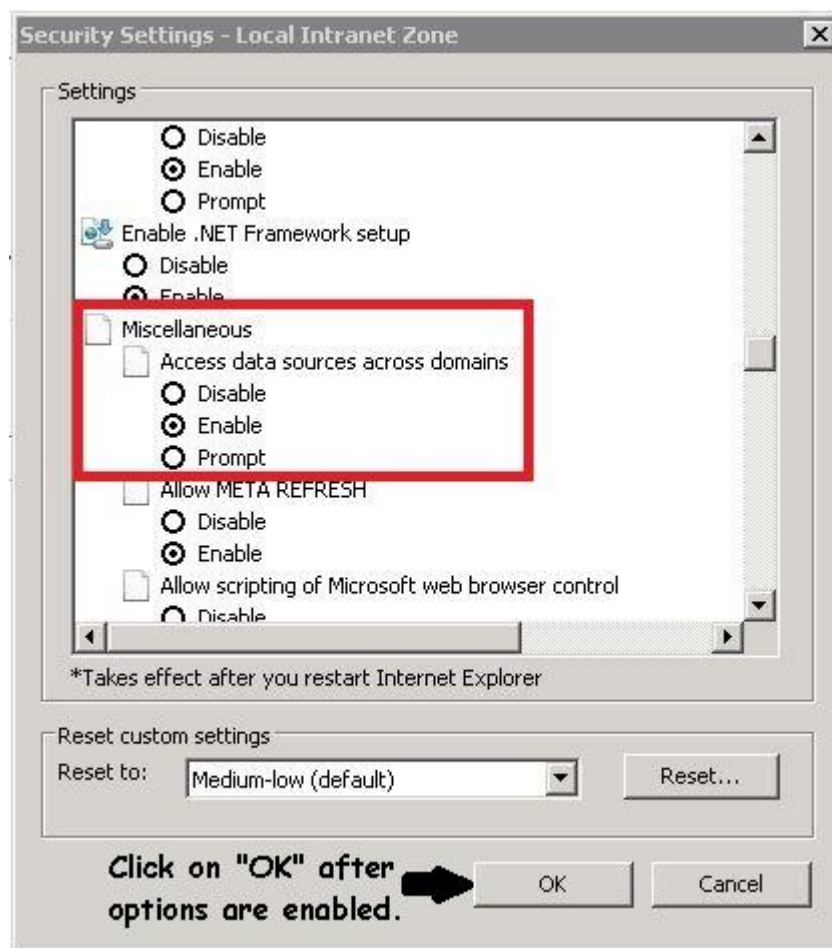


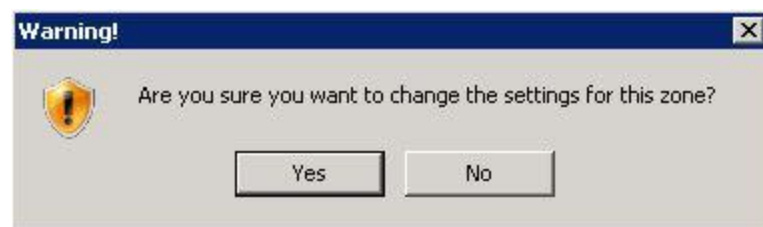


Select **Enable** up to Script ActiveX controls marked safe for scripting*

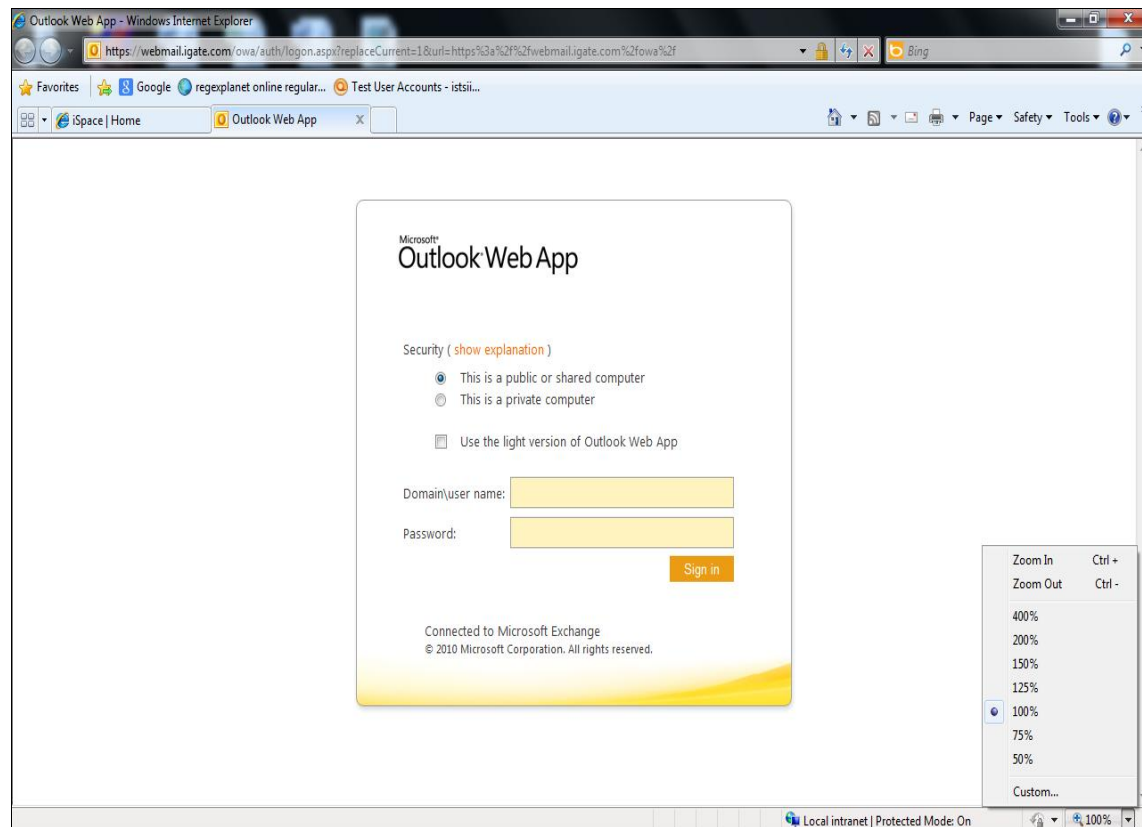


Step 4: Scroll down to Miscellaneous Options, Select **Enable** for "Access data sources across domains". **Step 4:** Click on "OK" button and then if it throws a warning message, click on Yes





Step 6: Make sure that Zoom is set to 100% in IE.



Step 7: Install IE.reg from the home folder. This is required to avoid unnecessary script prompts.



Section 4. Using Step Generator...

4.1 Prerequisites

- **HP Quality Center credentials** should be available.
- Browser setting should be set prior to launch the login file.

4.2 Login to Step Generator

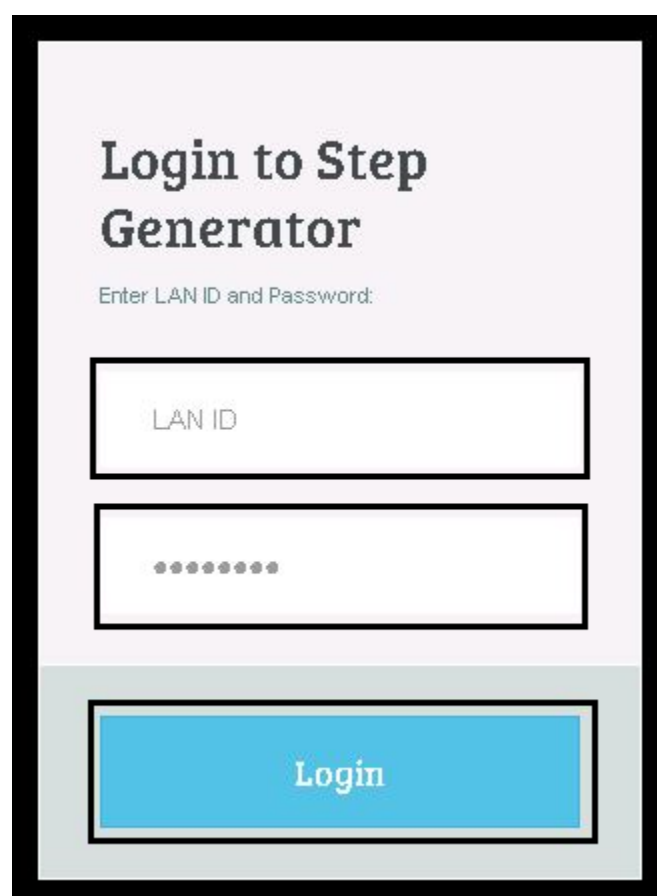
Step 1: Open 'Login.html' placed in the 'Step Generator' folder.



Step 2: Change the **config** settings if required from the link available on the Login Page (Right Bottom of the page).



Step 3: Enter credentials (Quality Center) and click on login button.



4.3 Application Maintenance

4.3.1 Select Application

This tab is used to select the application available from the list.

Application Name: Drop Down with all available applications.

Application OR Path: Not editable. This box shows the path of Application Object Repository Path.

XML Script Path: To select the existing Test file (in xml format).

The screenshot shows the 'Select Application' tab selected in a menu bar. Below the menu bar, the title 'Select Application:' is displayed in green. There are three input fields: 'Application Name' with a dropdown menu showing 'Compass', 'Application OR Path' with a text box containing 'T:\Test Services\Offshore Folder\Final_VI\HTML PROJ\XML\Application OR\Compass\Compass.xml', and 'XML Script Path' with a text box and a 'Browse...' button. A blue 'Continue' button is located to the right of the 'Application Name' field.

4.3.2 Add Application

This tab is used to add new application.

Step 1: On Step Generator Home Page, click on “Add Application” tab.

The screenshot shows the 'Add Application' tab selected in a menu bar. Below the menu bar, the title 'Add Application:' is displayed in green. There are two input fields: 'Application Name' with a text box and 'Application OR Path' with a text box. A blue 'Add Application' button is located to the right of the 'Application Name' field.

Step 2: Enter Name of the application and path of the Object Repository in XML format of the new application.

Example:

Application Name: “TestApplication”

Application OR Path: “T:\Test Folder\TestApplication.xml”

The screenshot shows the 'Add Application' tab with the example data entered. The 'Application Name' field contains 'TestApplication' and the 'Application OR Path' field contains 'T:\Test Folder\TestApplication.xml'. The blue 'Add Application' button is still present.

Step 3: Click on **Add Application** button. Click “OK” on the message box.



4.3.3 Edit Application

This tab is used to update the **Application OR Path**.

Step 1: On Step Generator Home Page, click on “**Update Application**” tab.

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Update Application:

Application Name:

Application OR Path:

Step 2: Enter Name of the application and path of the Object Repository in XML format of the new application.

Example:

Application Name: “TestApplication”

Application OR Path: “T:\TestApplication.xml”

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Update Application:

Application Name:

Application OR Path:

Step 3: Click on **Edit Application** button. Click “**OK**” on the message box.



4.3.4 Delete Application

This tab is used to delete the **Application**.

Step 1: On Step Generator Home Page, click on “**Update Application**” tab.

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Delete Application:

Application Name:

Application OR Path:

Step 2: Select Application from the list.

Example:

Application Name: "TestApplication"

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Delete Application:

Application Name:

Application OR Path:

Step 3: Click on **Delete Application** button. Click "OK" on the message box.



4.4 Object Repository Management

4.4.1 Adding Objects in Object Repository

Adding TestObject in object Repository

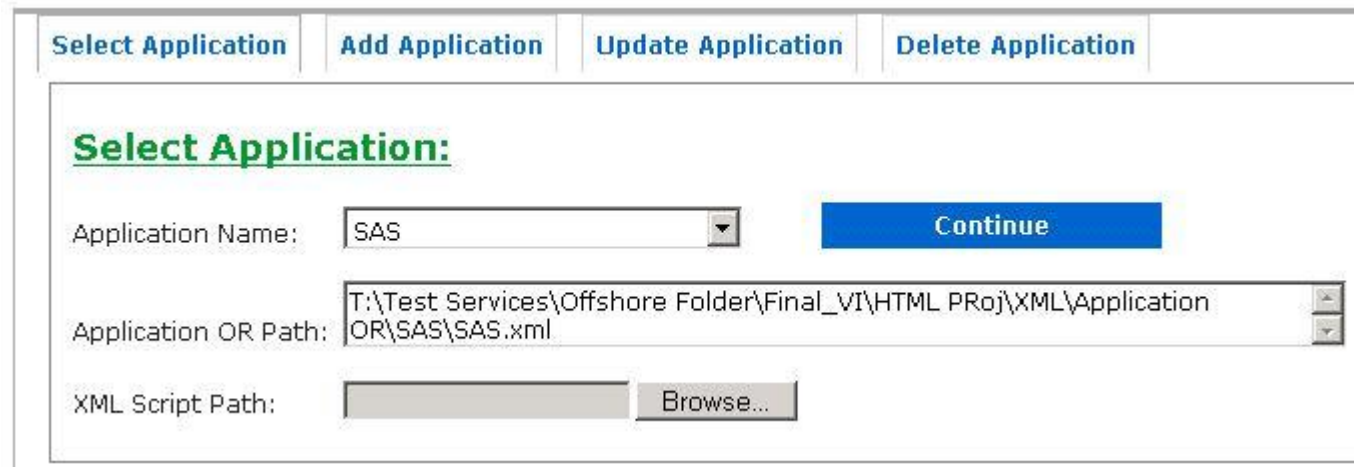
In this example, TestObject is WebEdit and its parent object is WebPage which is AEB_Page which in turn is child of Browser named AEB_Browser.

i.e. Hierarchy is like:

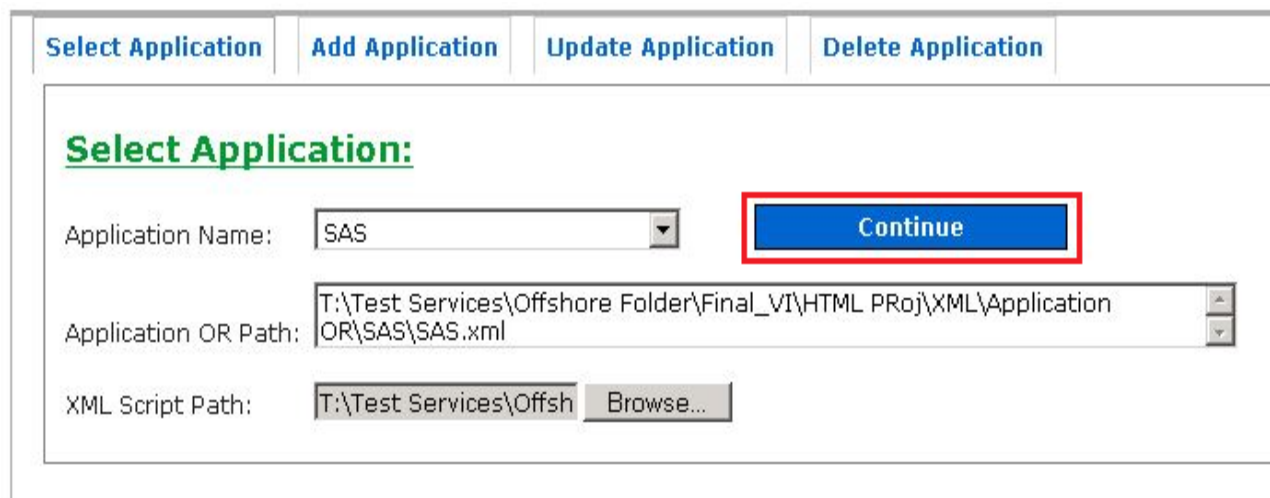
AEB_Browser().AEB_Page().TestObject()

Step 1: Select Application from the list.

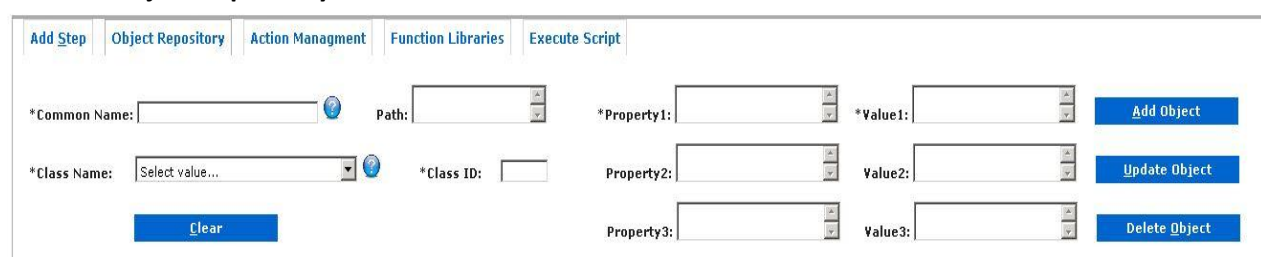
Example: SAS



Step 2: Click on “Continue” button.



Step 3: Click on “Object Repository” Tab.



Step 4: Enter Common Name; it should be unique and not used before.

Example: TestObject (WebEdit)

Enter Path. Hierarchy to parent objects.

Example: AEB_Browser.AEB_Page

Select Class Name, Class ID will be populated automatically.

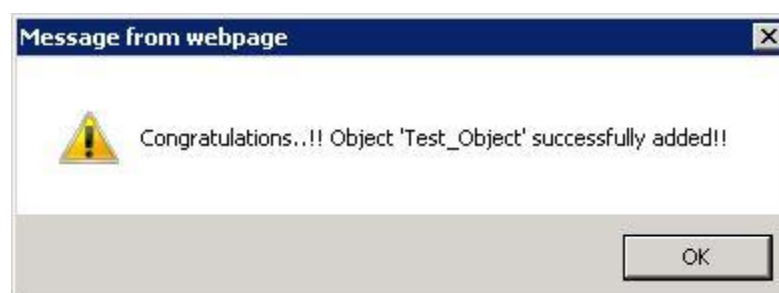
Example: WebEdit

Add Step	Object Repository	Action Management	Function Libraries	Execute Scri
*Common Name: <input type="text" value="TestObject"/>		Path: <input type="text" value="AEB_Browser.AEB_Page"/>		
*Class Name: <input type="text" value="WebEdit"/>		*Class ID: <input type="text" value="16"/>		
<input type="button" value="Clear"/>				

Step 5: Using QTP “Object Spy”, get the object properties and add them.

*Property1:	<input type="text" value="Property1"/>	*Value1:	<input type="text" value="Value1"/>
Property2:	<input type="text" value="Property2"/>	Value2:	<input type="text" value="Value2"/>
Property3:	<input type="text" value="Property3"/>	Value3:	<input type="text" value="Value3"/>

Step 6: Click on “Add Object”. Click on Ok Button on message box.



4.4.2 Editing Objects in Object Repository

Editing TestObject in object Repository

In this example, TestObject is WebEdit and its parent object is WebPage which is AEB_Page which in turn is child of Browser named AEB_Browser.

i.e. Hierarchy is like:

AEB_Browser().AEB_Page().TestObject()

Step 1: Select Application from the list.

Example: SAS

Select Application | Add Application | Update Application | Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 2: Click on “Continue” button.

Select Application | Add Application | Update Application | Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: T:\Test Services\Offsh Browse...

Step 3: Click on “Object Repository” Tab.

Add Step | Object Repository | Action Management | Function Libraries | Execute Script

*Common Name: Path: AEB_WriteUp *Property1: micclass *Value1: WebEdit Add Object

*Class Name: WebEdit *Class ID: 16 Property2: name Value2: Test Update Object

Property3: Value3: Delete Object

Clear

Step 4: Enter Common Name and press “Tab”, this will load the properties

Example: TestObject (WebEdit)

Add Step | Object Repository | Action Management | Function Libraries | Execute Script

*Common Name: Test_Object Path: AEB_WriteUp *Property1: micclass *Value1: WebEdit Add Object

*Class Name: WebEdit *Class ID: 16 Property2: name Value2: Test Update Object

Property3: Value3: Delete Object

Clear

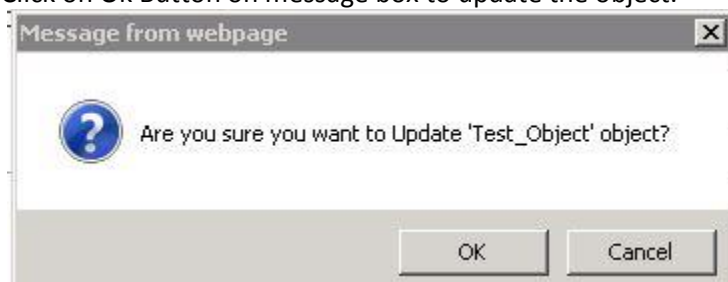
Step 5: Using QTP “**Object Spy**”, get the object properties and update the required ones or add new property if available. User can also update Path and Class.

*Property1:	<input type="text" value="micclass"/>	*Value1:	<input type="text" value="WebEdit"/>
Property2:	<input type="text" value="name"/>	Value2:	<input type="text" value="UpdatedTest"/>
Property3:	<input type="text" value="NewProperty"/>	Value3:	<input type="text" value="NewValue"/>

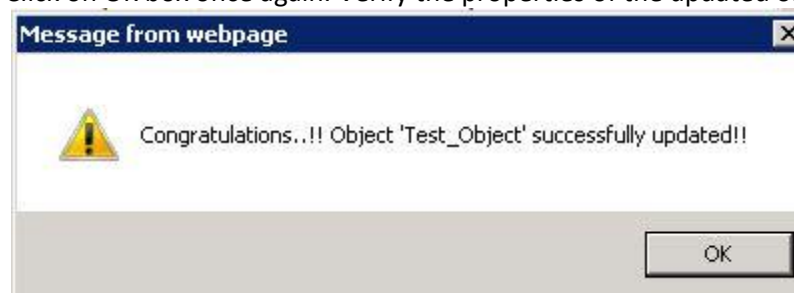
Step 6: Click on “**Update Object**”.



Step 7: Click on Ok Button on message box to update the object.



Step 8: Click on OK box once again. Verify the properties of the updated object.

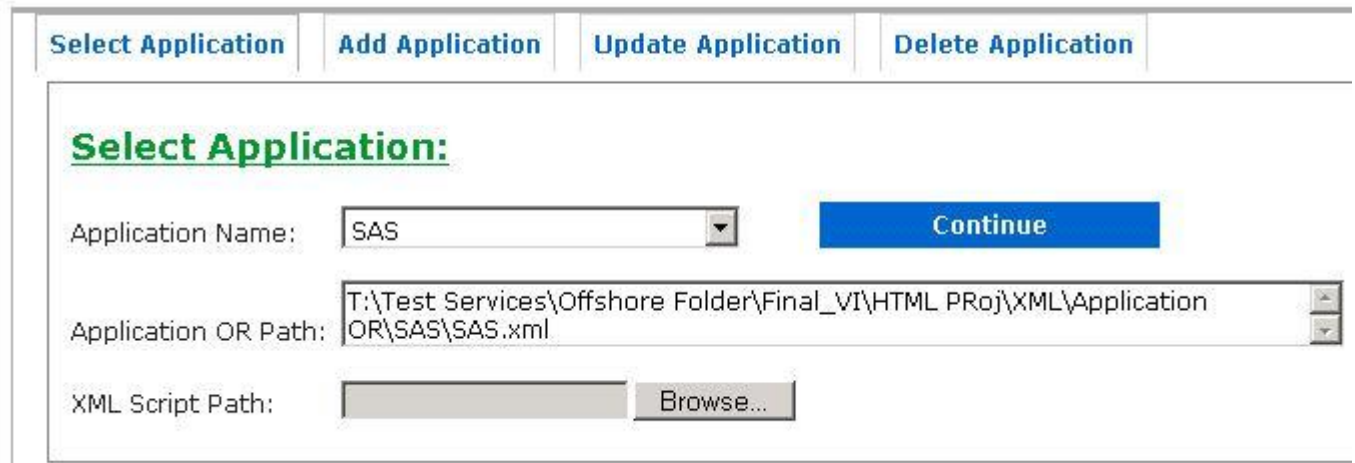


4.4.3 Deleting Objects in Object Repository

Deleting TestObject in object Repository

Step 1: Select Application from the list.

Example: SAS



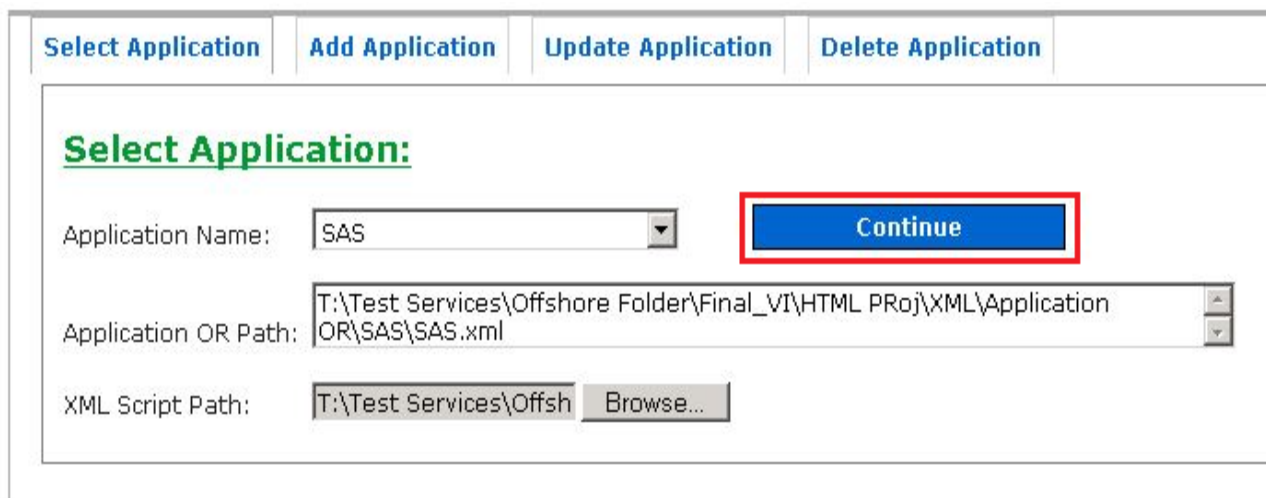
Select Application

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 2: Click on “Continue” button.



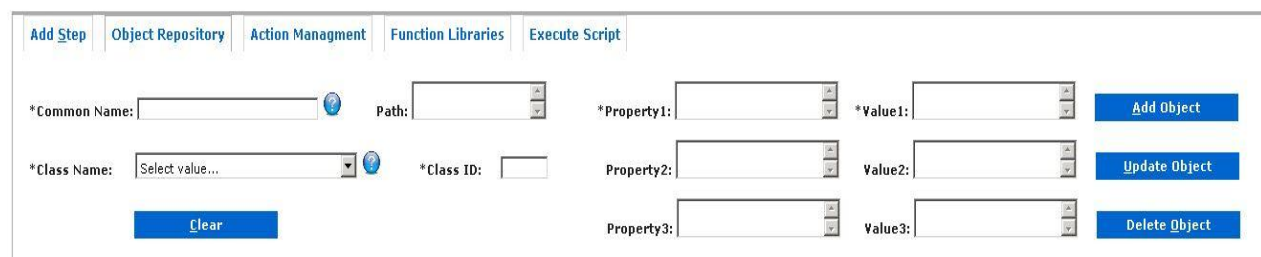
Select Application

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: T:\Test Services\Offsh Browse...

Step 3: Click on “Object Repository” Tab.



Add Step Object Repository Action Managment Function Libraries Execute Script

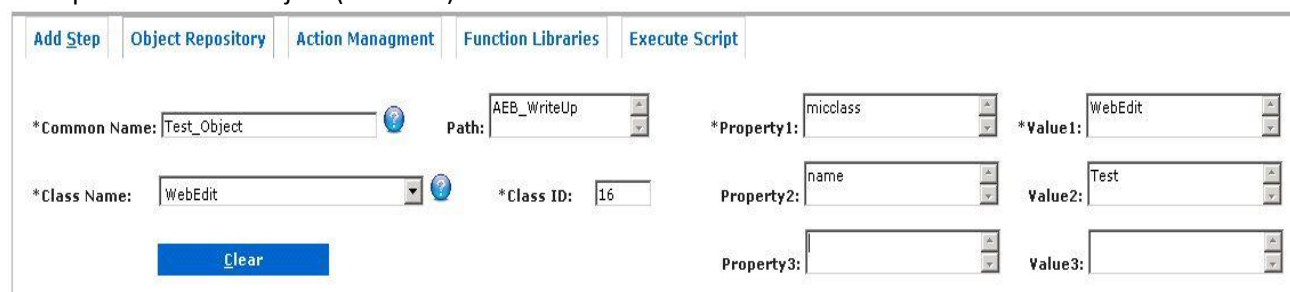
*Common Name: Path: *Property1: *Value1: Add Object

*Class Name: Select value... *Class ID: Property2: Value2: Update Object

Clear Property3: Value3: Delete Object

Step 4: Enter Common Name and press “Tab”, this will load the properties

Example: TestObject (WebEdit)



Add Step Object Repository Action Managment Function Libraries Execute Script

*Common Name: Test_Object Path: AEB_WriteUp *Property1: micclass *Value1: WebEdit

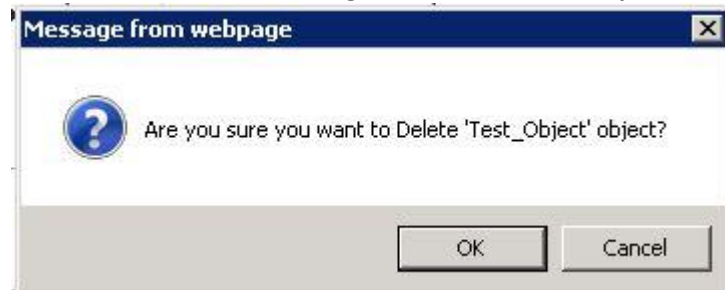
*Class Name: WebEdit *Class ID: 16 Property2: name Value2: Test

Clear Property3: Value3:

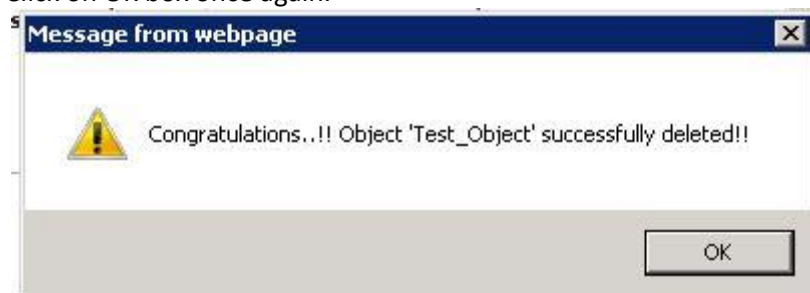
Step 5: Click on “Delete Object”.



Step 6: Click on Ok Button on message box to delete the object.



Step 7: Click on OK box once again.



4.5 Action Management

This tab is used to manage actions.

Whenever new keyword is added in the function library (in QTP) we need to add the corresponding action name also. This task is achieved from Action Management tab.

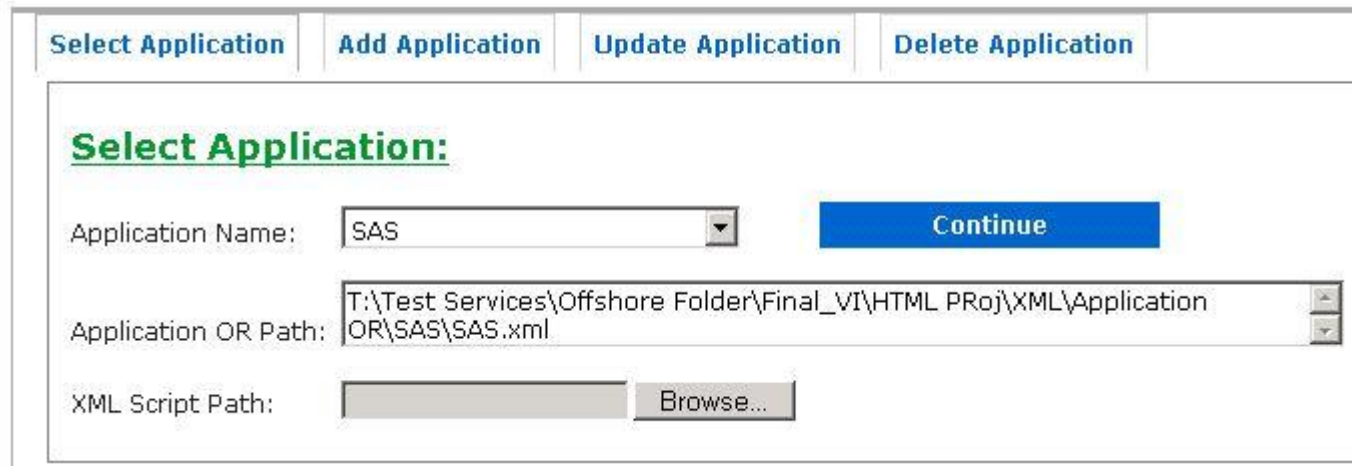
Also , sometimes we also need to update or delete the action, same thing can also be done from Action Management tab.

4.5.1 Adding Actions in Step Generator

Below are the steps for adding action in the utility.

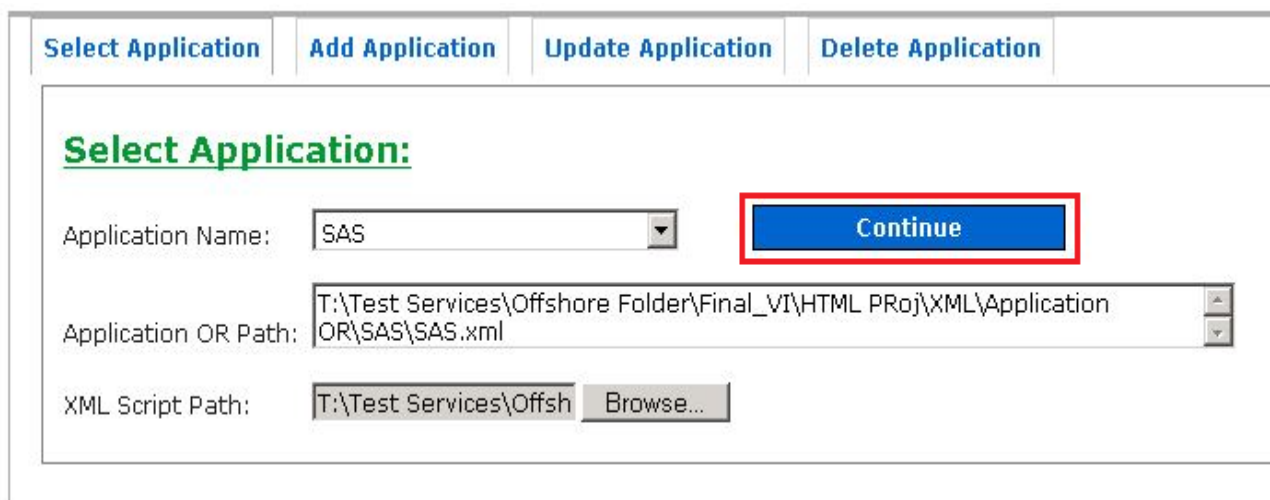
Step 1: Select Application from the list.

Example: SAS



The screenshot shows a dialog box titled "Select Application:". It has four tabs: "Select Application", "Add Application", "Update Application", and "Delete Application". The "Select Application" tab is active. Inside the tab, there is a form with the following fields: "Application Name:" with a dropdown menu showing "SAS"; a blue "Continue" button; "Application OR Path:" with a text box containing "T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml"; and "XML Script Path:" with an empty text box and a "Browse..." button.

Step 2: Click on "Continue" button.



This screenshot is identical to the previous one, but the "Continue" button is highlighted with a red rectangular border.

Step 3: Click on "Action Management" Tab.

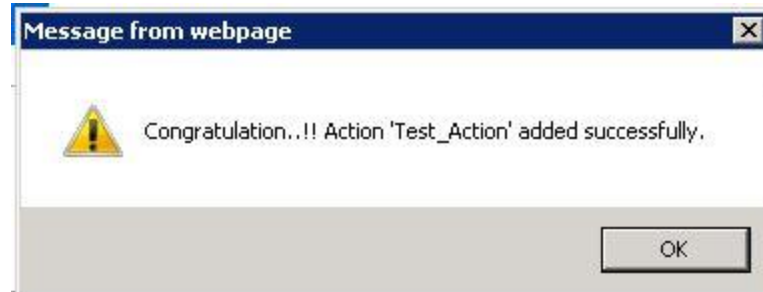


The screenshot shows the "Action Management" tab selected. The "Add Step" tab is also visible. The "Action Name" field is empty. The "Updated Action Name" field is empty. The "Add Action", "Update Action", and "Delete Action" buttons are visible.

Step 4: Enter the name of the action.
Example: "Test_Action"

The screenshot shows a web application interface with four tabs: 'Add Step', 'Object Repository', 'Action Management', and 'Function Librarie'. The 'Add Step' tab is active. Below the tabs, there is a text input field labeled 'Action Name:' containing the text 'Test_Action'. To the right of the input field is a blue circular icon with a question mark. Further right is the text 'Updated Act'. At the bottom of the interface are three blue buttons: 'Add Action', 'Update Action', and 'Delete ction'.

Step 5: Click on "Add Action" Button. Click on "OK" button on the alert box.



4.5.2 Editing Actions in Step Generator

Below are the steps for editing action in the utility.

Step 1: Select Application from the list.

Example: SAS

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 2: Click on “Continue” button.

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 3: Click on “Action Management” Tab.

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Action Name:

Updated Action Name:

Step 4: Enter the name of the action and **Updated Action Name** to change the name.

Example: Action Name: “Test_Action”
 Updated Action Name: “Test_Action_Updated”

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Action Name:

Updated Action Name:

Step 5: Click on “**Update Action**” Button. Click on “**OK**” button on the alert box.



4.5.3 Deleting Actions in Step Generator

Below are the steps for deleting action in the utility.

Step 1: Select Application from the list.

Example: SAS

Select Application Add Application Update Application Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 2: Click on “Continue” button.

Select Application Add Application Update Application Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: T:\Test Services\Offsh Browse...

Step 3: Click on “Action Management” Tab.

Add Step Object Repository Action Management Function Libraries Execute Script

Action Name: ? Updated Action Name:

Add Action Update Action Delete ction

Step 4: Enter the name of the action to delete.

Example: Action Name: “Test_Action_Updated”

Add Step Object Repository Action Management Function Librari

Action Name: Test_Action_Updated ? Updated Ac

Add Action Update Action Delete ction

Step 5: Click on “**Delete Action**” Button. Click on “**OK**” button on the alert box.



4.6 Function Library Management

This tab is used to manage Function Library.

Whenever new Function Library is added (in QTP) we need to add the corresponding Function Library name also. This task is achieved from Function Libraries tab.

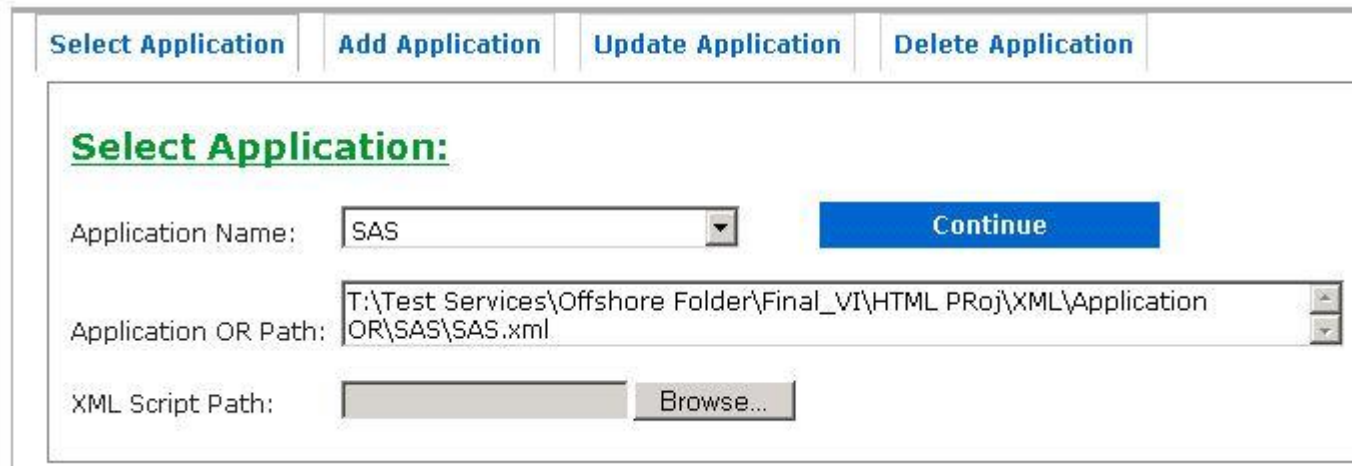
Also , sometimes we also need to update or delete the Function Library, same thing can also be done from Function Libraries tab.

4.6.1 Adding Function Library in Step Generator

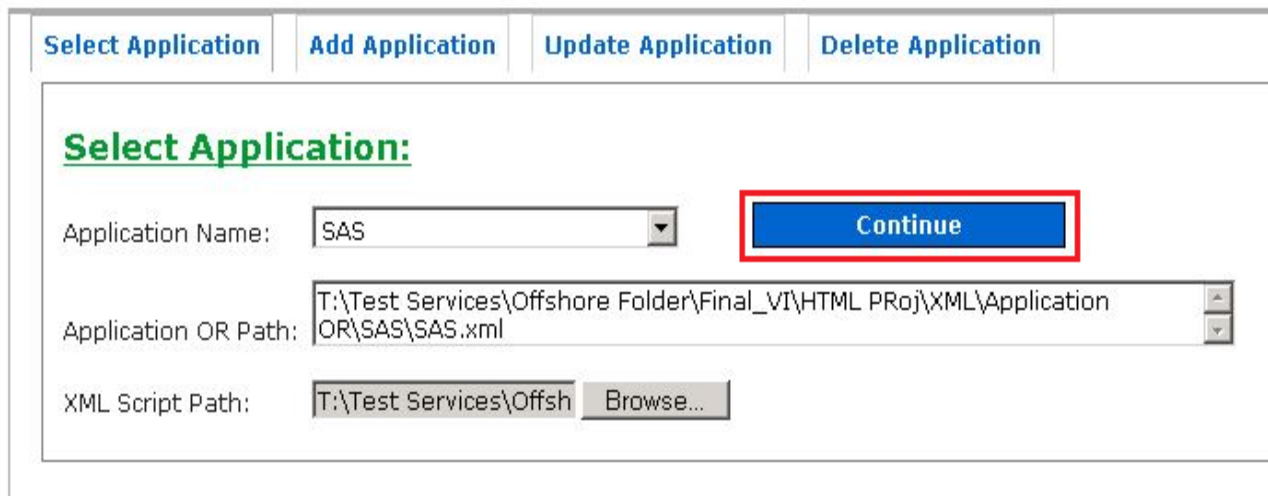
Below are the steps for adding Function Library in the utility.

Step 1: Select Application from the list.

Example: SAS



Step 2: Click on “Continue” button.



Step 3: Click on “Function Libraries” Tab.



Step 4: Enter the name of the Function Library.

Example: Function Library: "Test_Func_Lib"
 Function Library Path: "T:\Test.qfl"

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
Function Library: <input type="text" value="Test_Func_Lib"/>		Function Library Path: <input type="text" value="T:\Test.qfl"/>		
Add Function Library		Update Library	Delete Library	

Step 5: Click on "Add Function Library" Button. Click on "OK" button on the alert box.



4.6.2 Editing Function Library path in Step Generator

Below are the steps for editing action in the utility.

Step 1: Select Application from the list.

Example: SAS

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 2: Click on “Continue” button.

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 3: Click on “Function Libraries” Tab.

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Function Library:

Function Library Path:

Step 4: Select Function Name and Enter **Updated** Action Name to change the path.

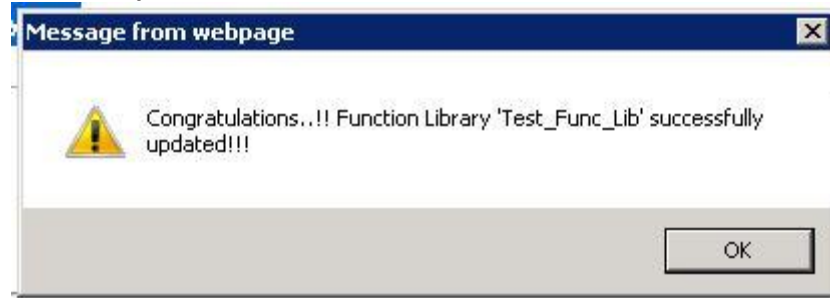
Example: Function Library: “Test_Action”
 Function Library Path: “T:\Test_Updated_Path.qfl”

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Function Library:

Function Library Path:

Step 5: Click on “**Update Action**” Button. Click on “**OK**” button on the alert box.



4.6.3 Deleting Function Library in Step Generator

Below are the steps for deleting Function Library in the utility.

Step 1: Select Application from the list.

Example: SAS

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 2: Click on “Continue” button.

Select Application	Add Application	Update Application	Delete Application
--------------------	-----------------	--------------------	--------------------

Select Application:

Application Name:

Application OR Path:

XML Script Path:

Step 3: Click on “Function Libraries” Tab.

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Function Library:

Function Library Path:

Step 4: Enter the name of the Function Library to delete.

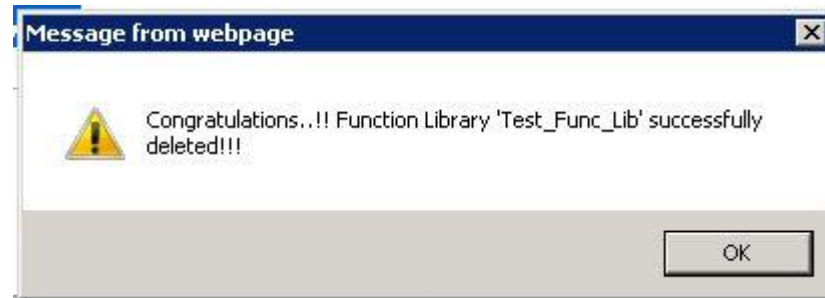
Example: Function Library: “Test_Func_Lib”

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
----------	-------------------	-------------------	--------------------	----------------

Function Library:

Function Library Path:

Step 5: Click on “**Delete Action**” Button. Click on “**OK**” button on the alert box.



4.7 Selecting existing Test Case (.xml)

Step 1: Select Application from the list.
Example: SAS

Select Application Add Application Update Application Delete Application

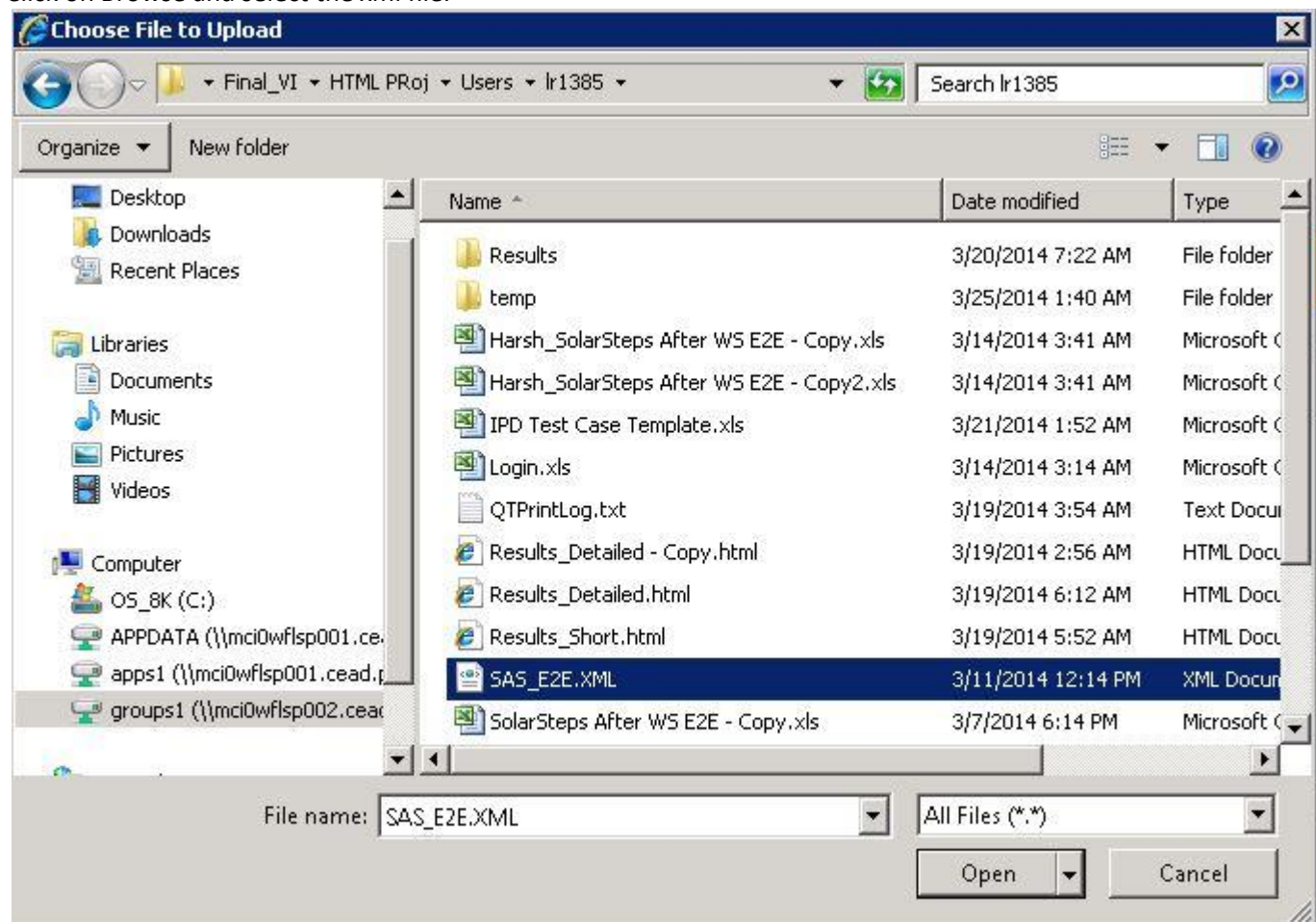
Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 2: Click on Browse and select the xml file.



Example: SAS_E2E.XML

NOTE: Only *.xml files are allowed to select.

Step 3: Click on "Continue" button.

Select Application Add Application Update Application Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml

XML Script Path: T:\Test Services\Offsh Browse...

Step 4: Verify Table in the bottom section of the Framework Utility Page.

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	SAS				func_startTest	"stage","TSAUTE2E","525d67cbe54748595a4688d86c6fcded41fa6f6e31713a0"	+ ADD
<input type="checkbox"/>	SAS			strGrpName	assign	"E2E_02242012"	+ ADD
<input type="checkbox"/>	SAS			strGroupOffice	assign	"Charlotte"	+ ADD
<input type="checkbox"/>	SAS			strProducerName	assign	"broker"	+ ADD
<input type="checkbox"/>	SAS			strProducerId	assign	"9000007900000"	+ ADD

4.8 Starting with new Test Case

Step 1: Select Application from the list.
Example: SAS

Select Application	Add Application	Update Application	Delete Application
Select Application:			
Application Name:	<input type="text" value="SAS"/>		<input type="button" value="Continue"/>
Application OR Path:	<input type="text" value="T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml"/>		
XML Script Path:	<input type="text" value=""/> <input type="button" value="Browse..."/>		

Step 2: Click on “Continue” button.

Select Application	Add Application	Update Application	Delete Application
Select Application:			
Application Name:	<input type="text" value="SAS"/>		<input type="button" value="Continue"/>
Application OR Path:	<input type="text" value="T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml"/>		
XML Script Path:	<input type="text" value="T:\Test Services\Offsh"/> <input type="button" value="Browse..."/>		

4.9 Add Step Tab

4.9.1 Writing Test Case

There are two ways of writing test cases using Step Generator

1. Using the GUI available in the top section.
2. Directly writing in the web-table.

4.9.1.1 Using the GUI

Understanding GUI:

The screenshot shows the 'Add Step' tab in the Step Generator GUI. It features several input fields: '*Test Case ID:', 'Skip:' (with a checkbox), 'Break:' (with a checkbox), 'Object:', '*Actions:', and 'Data:'. There are also help icons (question marks) next to the 'Object:' and 'Actions:' fields. At the bottom, there is a row of buttons: 'Add Step', 'Delete Row', 'Clear', 'Export To Excel', 'Save Script', and 'Save As'.

Field Name	Description
Test Case ID	Mandatory field. Test Case ID of the test Script Example "TC01"
Skip	Check Box. Check to skip the step.
Break	Check Box. Check to have a break during execution on the step (For debugging).
Object	Textbox. Object Name. Fetched from object repository.
Actions	Mandatory field. Action Name or "keyword".
Data	Textbox. For entering data of the step.

Below are the steps for writing test case.

Step 1: Select Application from the list.

Example: SAS

The screenshot shows the 'Select Application' dialog box. It has a title bar with buttons: 'Select Application', 'Add Application', 'Update Application', and 'Delete Application'. The main area has the title 'Select Application:' in green. Below it are three input fields: 'Application Name:' with 'SAS' selected, 'Application OR Path:' with 'T:\Test Services\Offshore Folder\Final_VI\HTML Proj\XML\Application OR\SAS\SAS.xml', and 'XML Script Path:' with a 'Browse...' button. A blue 'Continue' button is also present.

Step 2: Click on “Continue” button.

Select Application Add Application Update Application Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 3: Enter details in Add Step Tab for

- ⇒ Test Case ID : “TC01”
- ⇒ Skip: “Uncheck”
- ⇒ Break: “Check”
- ⇒ Object: “AEB_WriteUp”
- ⇒ Action: “msgbox”
- ⇒ Data: “This is a test step”

Add Step Object Repository Action Management Function Libraries Execute Script

*Test Case ID: TC01 Skip: ☐ Break: ☒ Object: AEB_WriteUp

*Actions: msgbox Data: This is a test step

Add Step Delete Row Clear Export To Excel Save

Step 4: Click on “Add Step” button. Verify Step in the Table below the tab.

	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	y	AEB_WriteUp	msgbox	This is a test step.	+ ADD

4.9.1.2 Using the Web-Table

We can directly use the web table (Only after first step is added from the GUI).
Below are the steps for writing test case.

Step 1: Select Application from the list.
Example: SAS

Select Application Add Application Update Application Delete Application

Select Application:

Application Name: SAS Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\XML\Application OR\SAS\SAS.xml

XML Script Path: Browse...

Step 2: Click on “Continue” button.

Select Application

Add Application

Update Application

Delete Application

Select Application:

Application Name: SAS

Continue

Application OR Path: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\XML\Application OR\SAS\SAS.xml

XML Script Path:

Browse...

Step 3: Enter details in Add Step Tab for

- ⇒ Test Case ID : “TC01”
- ⇒ Skip: “Uncheck”
- ⇒ Break: “Check”
- ⇒ Object: “AEB_WriteUp”
- ⇒ Action: “msgbox”
- ⇒ Data: “This is a test step”

Add Step

Object Repository

Action Management

Function Libraries

Execute Script

*Test Case ID: TC01

Skip: ☐ Break: ☒ Object: AEB_WriteUp

*Actions: msgbox

Data: This is a test step

Add Step

Delete Row

Clear

Export To Excel

Save

Step 4: Click on “Add Step” button. Verify Step in the Table below the tab.

Add Step

Delete Row

Clear

Export To Excel

Save Script

SaveAs

	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AEB_WriteUp	msgbox	This is a test step.	+ ADD

Step 5: Click on “+ Add” button in the last column of the Web-Table. Test Case will automatically be populated.

	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AEB_WriteUp	msgbox	This is a test step.	+ ADD
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>				+ ADD

Step 6: Enter the details same as required.

- ⇒ Test Case ID : “TC01”
- ⇒ Skip: “Uncheck”
- ⇒ Break: “Uncheck”
- ⇒ Object: “AEB_WriteUp”
- ⇒ Action: “msgbox”
- ⇒ Data: “This is a second test step”

	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AEB_WriteUp	msgbox	This is a test step.	+ ADD
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		msgbox	This is a second test step	+ ADD

4.9.1.3 Re-Usable Scripts

Step Generator can also handle Re-usable scripts.

Syntax: **mod_**

Use mod_ in action to along with some prefix (for eg: mod_Test1, mod_Test2 etc.)

In Data column enter the path of the xls file you want to use.

For example: If Login is the file (**Location:** T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Login.xls) which we want to use in our Test Script then we will write:

Action: mod_Login

Data: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Login.xls

Similarly if we want to use Logout File (**Location:** T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Logout.xls) than we will write:

Action: mod_Logout

Data: T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Logout.xls

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		msgbox	hi	+ ADD
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		mod_Login	T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Login.xls	+ ADD
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		mod_Logout	T:\Test Services\Offshore Folder\Final_VI\HTML PRoj\Users\lr1385\Harsh Temp\Logout.xls	+ ADD
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		msgbox	bye	+ ADD

4.9.1.4 User Defined Functions

For using function we have a keyword/ prefix called **func_**

Similar to mod functionality, for function also we will add "**func_**" as the prefix before the function name. Like: func_FunctionName.

In Data we will write the input parameters for the function

"Parameter1", "Paramter2"

Suppose we want to use a custom function named "FormatDateinMMDDYYYYFormat" in our test case then we will write:

Action: func_FormatDateinMMDDYYYYFormat

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	SAS	<input type="checkbox"/>	<input type="checkbox"/>		func_startTest	"stage", "TSAUTE2E", "525d67cbe5474f8595a4688d86c6fcded41fa6f6e31713a0"	+ ADD

4.9.2 Deleting Test Steps

Deleting test steps can be performed on:

1. Individual Step
2. Multiple random steps
3. All the test steps.

User need to click “Delete” button.

4.9.2.1 Deleting Single Test Step

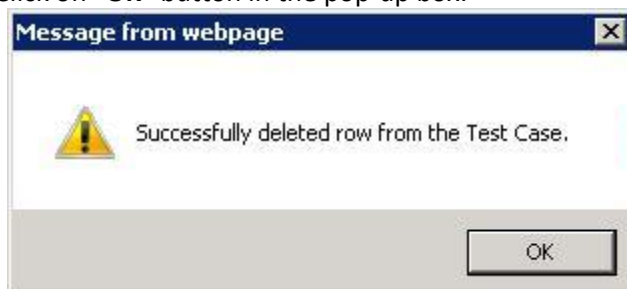
Step 1: Mark/Check the checkbox in first column

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>		comment	This is first step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is second step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is first delete step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fifth step.	+ ADD

Step2: Click on “Delete Row” button.



Step3: Click on “OK” button in the pop-up box.



Step4: Step is deleted:

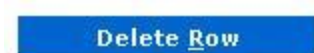
<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>		comment	This is first step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is second step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fifth step.	+ ADD

4.9.2.2 Deleting multiple steps from Test Case

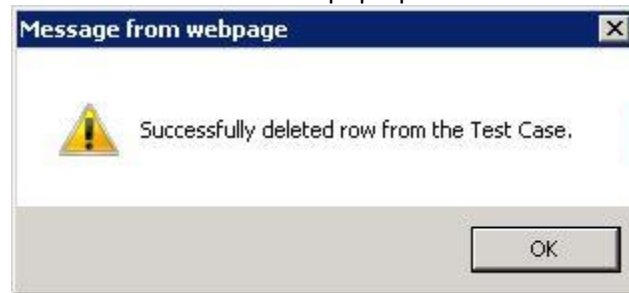
Step 1: Mark/Check the checkbox in first column against the steps to delete

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>		comment	This is first step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is first delete step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is second delete step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is sixth step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is seventh step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is third delete step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth delete step.	+ ADD

Step2: Click on “Delete Row” button.



Step3: Click on “OK” button in the pop-up box.



Step4: Mark steps are deleted:

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>		comment	This is first step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is sixth step.	+ ADD
<input type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is seventh step.	+ ADD

4.9.2.3 Deleting all steps from Test Case

Step 1: Mark/Check the checkbox in header row.

<input checked="" type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>		comment	This is first step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is fourth step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is sixth step.	+ ADD
<input checked="" type="checkbox"/>	TC001	<input type="checkbox"/>	<input type="checkbox"/>			This is seventh step.	+ ADD

Step2: Click on “Delete Row” button.



Step3: Click on “OK” button in the pop-up box.



Step4: Mark steps are deleted:

<input type="checkbox"/>	TestCase	Skip	Break	Object	Action	Data	Add Step
--------------------------	----------	------	-------	--------	--------	------	----------

4.9.3 Saving Test Case

Saving xml can be done in two ways:

4. Saving to temp File (Save Script)
5. Saving to custom folder (Save As Script)



4.9.3.1 Saving to temp File

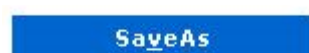
Click on **“Save Script”** button to save directly to the temp file which is saved in the temp folder of the logged in user.



4.9.3.2 Saving to custom folder

Steps to save the script to custom folder.

Step 1: Click on **“SaveAs”** button.



Step2: Enter the path to save file. Click on **“OK”** button.



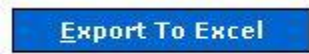
Step3: Click on **“OK”** button in the pop-up box.



4.9.4 Exporting Test Case (In Excel)

Steps to export test script in to excel.

Step 1: Click on “**Export To Excel**” button.



Step 2: Enter the path to save file. Click on “**OK**” button.



Step3: Wait for pop-up box and click on “**OK**” button in the pop-up box.



4.10 Execute Script Tab

4.10.1 Understanding Execute Script Tab

Add Step

Object Repository

Action Management

Function Libraries

Execute Script

Enter XLS path:

Browse...

[Click here to preview Excel File](#)

Enter Test Id:

Exit QTP after execution:

No

☒ Yes

☐

Function Library:

AEB

IPD

OA

SAS

Solar

Select Add-ins:

☐ ActiveX

☐ Terminal Emulators

☐ Java

☐ Visual Basic

☐ Oracle

☒ Web

☐ Stingray

☐ GWT

Execute Script

Last Run Result:

Field Name	Description
Enter XLS Path	This field is optional. If user wants to execute any excel test case he can execute by selecting excel file.
Click here to preview Excel File	Link which gives preview of the selected excel file in “Enter XLS Path” field.
Enter Test ID	Mandatory Field. It will execute the rows having the same value as mentioned in the box. Example: TC01
Exit QTP after execution	Radio Button. Determines whether to exit QTP or not after execution.
Exit QTP after execution	Mandatory field. Action Name or “ keyword ”.
Function Library	Web List. Select the function libraries that are applicable for the test script execution.
Last Run Result	Shows the result for the last run. Possible values: Fail, Pass, Stopped, Done.
Select Add-ins	Checkboxes. Allows user to select the QTP Add-ins which are applicable for the particular test script.

4.10.2 Executing Script

Scripts can be executed from utility in two ways:

1. Selecting the Excel script.
2. Executing the script written from the step generator.

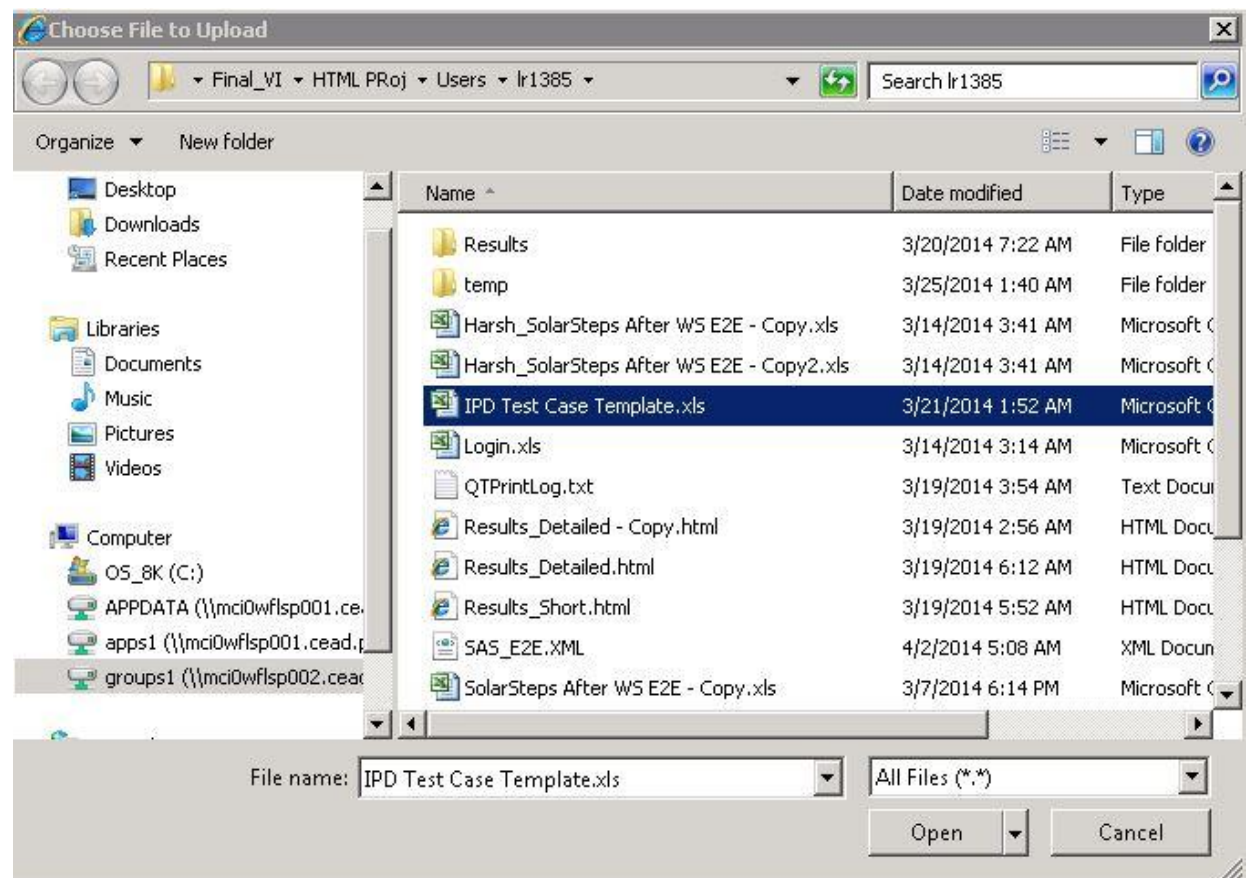
Mandatory Fields:

- ⇒ Test Case ID
- ⇒ QTP Add-ins
- ⇒ Function Library if applicable.

4.10.2.1 Selecting the Excel file

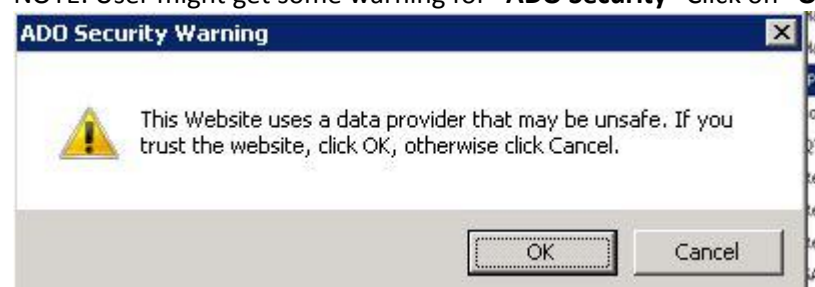
User can select Excel file and can execute the script.

Step 1: Click on “Browse” button and select the excel file.



Step 2: Click on “Click here to preview Excel File” link to cross verify if the selected file is correct or not.

NOTE: User might get some warning for “ADO Security” Click on “OK” button.



Once the file is loaded the preview will be like:

Test Case Preview - Windows Internet Explorer

Test Case Preview: t:/test services/offshore folder/final_vi/html proj/users/lr1385/ipd test case template.xls

TestCase ID	Skip	Break	Object	Action	Data
PPDSingleALcc				comment	Enroll Individual - Alabama
PPDSingleALcc				func_startTest	"stage", "Alabama"
PPDSingleALcc			individualRadioButton	select	#0
PPDSingleALcc			firstName	enter	Robert
PPDSingleALcc			lastName	enter	Lee
PPDSingleALcc			genderMale	select	#1
PPDSingleALcc			dobMonth	select	Aug
PPDSingleALcc			dobDay	select	22
PPDSingleALcc			dobYear	select	1978
PPDSingleALcc			ssnChunk1	enter	[randomssn1]
PPDSingleALcc			ssnChunk2	enter	[randomssn2]
PPDSingleALcc			ssnChunk3	enter	[randomssn3]

Step 3: Close the preview window and

Enter Test Case ID:

Example: **PPDSingleALcc**

Select Function Libraries:

Example: **IPD**

Select Add-ins:

Example: **Web**

Add Step | Object Repository | Action Management | Function Libraries | Execute Script

Enter XLS path: T:\Test Services\Offsho Browse...

Click here to preview Excel File

Enter Test ID: PPDSingleALcc

Exit QTP after execution: No Yes

Function Library: AEB IPD OA SAS Solar

Select Add-ins: ☐ ActiveX ☐ Terminal Emulators ☐ Java ☐ Visual Basic ☐ Oracle ☒ Web ☐ Stingray ☐ GWT

Execute Script

Last Run Result:

Step 4: Click on “Execute Script” button.

Wait for QTP to complete the execution.

Step 5: Get the **Last Run Result**.

Last Run Result: Passed

4.10.2.2 Executing after writing script in Utility

User can execute the script which he/she has written in the utility

Step 1: Write test steps using Utility.

	TestCase	Skip	Break	Object	Action	Data	Add Step
<input type="checkbox"/>	TC01	<input type="checkbox"/>	<input type="checkbox"/>		msgbox	This is a test execution	+ ADD

Step 2: Enter details for:

Enter Test Case ID:

Example: **TC01**

Select Function Libraries:

Example: **IPD**

Select Add-ins:

Example: **Web**

Add Step	Object Repository	Action Management	Function Libraries	Execute Script
<div>Enter XLS path: <input type="text"/> Click here to preview Excel File <input type="button" value="Browse..."/></div> <div>Enter Test ID: <input type="text" value="TC01"/></div> <div>Exit QTP after execution: No <input checked="" type="radio"/> Yes <input type="radio"/></div> <div><input type="button" value="Execute Script"/></div> <div>Function Library: <div><div>AEB</div><div>IPD</div><div>OA</div><div>SAS</div><div>Solar</div></div></div> <div>Select Add-ins:<div><div><input type="checkbox"/> ActiveX</div><div><input type="checkbox"/> Java</div><div><input type="checkbox"/> Oracle</div><div><input type="checkbox"/> Stingray</div></div><div><div><input type="checkbox"/> Terminal Emulators</div><div><input type="checkbox"/> Visual Basic</div><div><input checked="" type="checkbox"/> Web</div><div><input type="checkbox"/> GWT</div></div></div> <div>Last Run Result:</div>				

Step 3: Click on “Execute Script” button.

Wait for QTP to complete the execution.

Step 4: Get the **Last Run Result**.

Last Run Result: **Passed**

4.10.3 Analyzing results

User can analyze the results by clicking on the status of the result i.e. “Fail”, “Passed”, etc.
Result will be opened in a new window.

Click on result status:

Last Run Result: Passed

New Window with detailed result:

Execution Report

Step	Status
Setting up the object dictionary.	Pass
Objects Added.	Pass
3/20/2014 6:27:00 AM ExecuteLine , comment, Enroll Individual - Alabama	Pass
3/20/2014 6:27:00 AM ExecuteLine , func_startTest, "stage", "Alabama"	Pass
3/20/2014 6:27:00 AM Calling custom function: startTest ""stage", "Alabama""	Pass
3/20/2014 6:27:34 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebRadioGroup("html id=gwt-debug-individualRadioButton-input", "index:=0"), select, #0	Pass
3/20/2014 6:27:34 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-firstName"), enter, Robert	Pass
3/20/2014 6:27:35 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-lastName"), enter, Lee	Pass
3/20/2014 6:27:35 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebRadioGroup("html id=gwt-debug-genderMale-input", "index:=0"), select, #1	Pass
3/20/2014 6:27:36 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebList("html id=gwt-debug-dobMonth"), select, Aug	Pass
3/20/2014 6:27:37 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebList("html id=gwt-debug-dobDay"), select, 22	Pass
3/20/2014 6:27:37 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebList("html id=gwt-debug-dobYear"), select, 1978	Pass
3/20/2014 6:27:38 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-ssnChunk1"), enter, [randomssn1]	Pass
3/20/2014 6:27:39 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-ssnChunk2"), enter, [randomssn2]	Pass
3/20/2014 6:27:39 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-ssnChunk3"), enter, [randomssn3]	Pass
3/20/2014 6:27:40 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-address"), enter, 2010 East Cedar Avenue	Pass
3/20/2014 6:27:41 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-city"), enter, Mobile	Pass
3/20/2014 6:27:41 AM ExecuteLine Browser("creationtime:=0").Page("title:=Individual PPD").WebEdit("html id=gwt-debug-city"), enter, Mobile	Pass