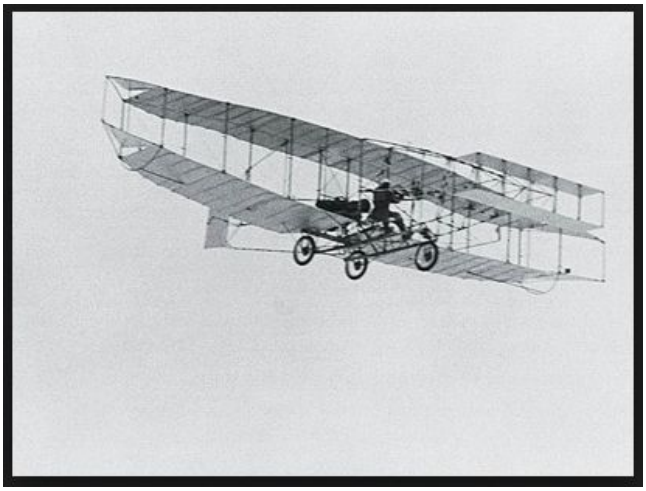
Chapter -7

UFT in Practical World



In this playhouse of infinite forms I have had my play, and here have I caught sight of him that is formless.

- Rabindranath Tagore (from *Gitanjali)*

In this chapter we will discuss the automation implementation in real world and UFT features to design that. We will discuss Checkpoints, Utility Objects, and Environment variable and reusable Library in this chapter

1.0 Software Test Automation Ingredient

Testing is an activity to verify the expected behaviour of a test condition with actual behaviour of application during run time. Either humans or computer programs can run these test steps. Human manually verify the actual outcome while programs need to explicitly check the value by implementing logic. UFT provides great facility to check the test steps. Testing also involve to record various other thing during execution e.g. record any information in report, generate random numbers and system and internet settings during execution.

Sometime we need system settings, Test properties and user input during test execution. UFT provides environment-variable concept to get test, system and user data during test execution. Actions modularize the test in reusable actions but action mainly used to record application test steps. User need to write logic to enhance the test e.g. creating Excel connection, reading text files, closing the process. We can write reusable function in VBScript library and they can be used in any test.

2.0 Checking the value in UFT

Testing is all about making verification (also known as assertions) during runtime. Tester may need to verify the page text, objects on page, Images, data in table in web page, data in table in database, properties of object etc. we can achieve verification in two ways

1. Checkpoints provide by UFT
2. VBScript logic to verify the condition

UFT Checkpoint are easiest way to verify the run-time object. Checkpoint can be inserted during record time or from "active screen" after recording the test.

Exercise 1.0 (of 2 STEPS) – Understanding the Active Screen

STEP1: Record a Test to search UFT "CheckPointTest" on Google

☝ *Navigate to UFT | New | Test | GUI Test | CheckPointTest | Click Create | Record | Select radio button* as in figure | http://Google.com | OK | Search with "Krishna wikipedia" | Search button | Click on Images Link | Click on Search Link | Click on First Link of Google Search | Close Internet Explorer | Save test*

*Recap – UFT provides Record and Run Settings. If user wants to run on specific website than option "Open the following address when a record or run session begin" should be selected. In this case browser automatically open with specified URL. It also provide options to skip recording of already open browser and close browser after run. Option "Record and run test on any open browser" will simply record test on any URL and browser.

STEP2: Let's analyse the recording. Open the active screen

Navigate to UFT | View | Active Screen | Right Click on Active Screen

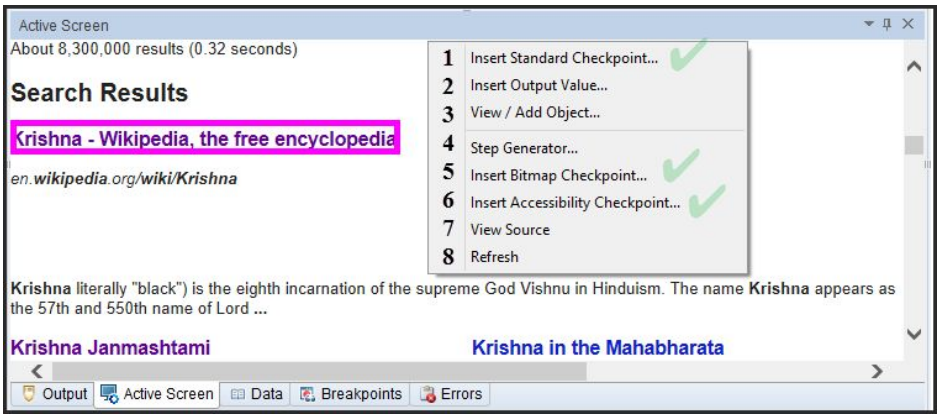
ConfuSense – The Active Screen provides a snapshot of your application as it appeared when you performed the corresponding step during a recording session. An Active Screen can be captured for every step you record. Additionally, depending on the Active Screen capture options that you used while recording, the page displayed in the Active Screen can contain detailed property information on each object displayed on the page.

Commented [ks1]: HP Guide

- HP Unified Functional Testing User Guide

Concept of Exercise – Understanding the Active Screen

In STEP1, we have recorded a test to search "Krishna wikipedia". In STEP2, we are exploring the active screen. When we record the test, UFT capture the screen shots known as "active screen". These are not just the screen shots but also enable tester to insert checkpoint, add object and view source. They are very helpful to modify and understand the test application.



Active screen allows to modify script even when application is not available. It provides features to add checkpoints (i.e. assert verifications points), View or add object (even not captured in OR), insert output value step to take the object property value from application. Active screen also provides other features to add steps in script and view page source.

Output value returns the object property value. Suppose there is a country dropdown with 200 country names. Output value enables UFT to capture all the 200 names of dropdown "Item" property and send it in to run-time datatable. It proved very helpful to write logic where we need to capture the dynamic value (e.g. amazon order number, price detail, dates etc.) during run-time and process it in our script logic for further navigation. Checkpoint returns true and false for property value while Output value returns the property value

itself. Checkpoint and Output value can be inserted during recording time or via active screen.

Hey hang on! Have you heard that? Aadhya's voice, asking for something in loud voice. Let's see what happened



Aric: Hey Aadhya, What happened. Why you are fighting in this room? What you are searching?

Aadhya: Aric, I am very upset. I have done manual testing on Google search for "Krishna Wikipedia". I made entries in notebook every time I ran the test. I verified the following on page

1. Link "Krishna - Wikipedia", Google Search box and other elements are present on page (Standard Checkpoint)
2. Image is appearing correctly (Image and Bit Map Checkpoint)
3. Text after search is appearing properly (Text Checkpoint)
4. Google website compliance with W3C standard for differently-able people (Accessibility Checkpoint)
5. Clicking on links are not leading to "404 Page not found" error (Page Checkpoint)

You see Aric! I ran that test 1000 times over 6 month period and now I missed that notebook. It took 20 minute to capture all these verifications. I just lost 20000 minutes or 333 hours or 13 working days just for one test. I am searching that notebook. Can you make UFT to generate or search the same notebook for me with all that entry?

Aric: Sorry Aadhya, I am not able to help you. Neither UFT, Mr. Robot, forged the documents nor do any manual task like searching your notebook. But UFT can do all your verification in 30 second i.e. 40 times less and keep result safe for future verifications. So, effectively your 13 days task can be done in 3.3 hours only if you have met with UFT before.

Aadhya: Are you joking. I know UFT can run the script for n-number of times but how it verify all these things?

Aric: Yes, they can. Let's first look the UFT checkpoint types.

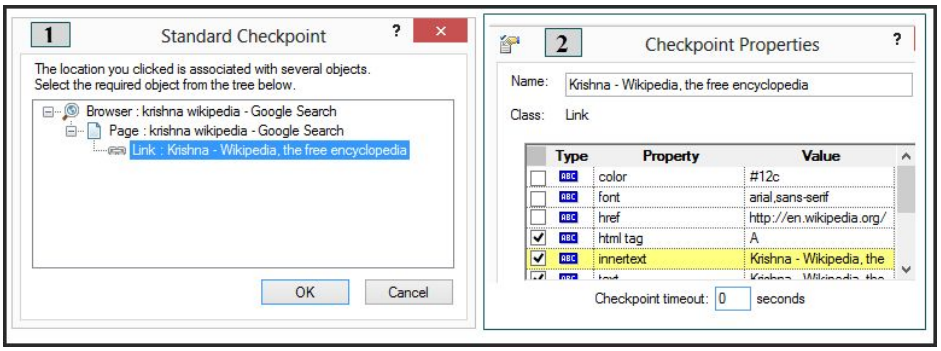
2.1 Checkpoint in UFT

Checkpoint can be inserted either at recording the test or from Active screen after recording.

Exercise 2.0 (of 3 STEPS) - Insert Checkpoint in Test Script

STEP1: Record a Test to search "Krishna Wikipedia" in Google and Insert page and link checkpoints

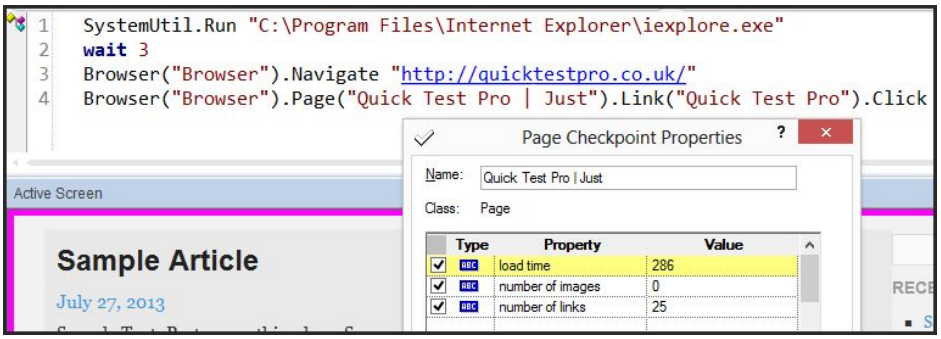
Navigate to UFT | New | Test | GUI Test |InsertCheckPointTest | Click Create | Record | Select first radio button in web tab| http://Google.com | OK | Search with "Krishna wikipedia" | Search button | Click insert step button |click Standard Checkpoint | Click wiki Link in google | Select "Link: Krishna – Wikipedia, the free encyclopedia" | uncheck href | keep other properties | OK | Click on Images Link | Click on Search Link | Close Browser | Stop recording | Save | Run by F5 key| View | Run Last Run Results | ALT + V + X | Close



STEP2: Insert Page checkpoint* by Active screen

Navigate to UFT | New | Test | GUI Test |PageCheckPointTest | Click Create | Record | Select first radio button in web tab| http://QuictestPro.co.uk | Click on any link | Stop | Save | View | Active screen | right click on Active screen | Insert Standard Checkpoint… | Select Page object | OK | OK | Verify the following line adds to script

```
5   Browser("Browser").Page("Quick Test Pro | Just").Check CheckPoint("Quick Test Pro | Just")
```



STEP3: Run test to check the page checkpoint added by active screen*

Navigate to "PageCheckPointTest" test in UFT | Run by F5 key | View | Run Last Run Results | ALT + V + X | Verify checkpoint | Close

ConfuSense – There are many checkpoints types in UFT. You will only get Standard, Accessibility and Bitmap checkpoint options in active screen. It does not mean other checkpoint scare from you. When you insert checkpoint on any object by standard checkpoint UFT will understand checkpoint type e.g. Image checkpoint will insert for Image object while text checkpoint will insert for label implicitly.

Concept of Exercise 2.0 - Insert Checkpoint in Test Script

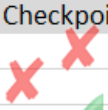
In STEP1, we are creating the standard checkpoint for link. Here we are getting opportunity to select multiple properties to check during run-time. When we insert checkpoint then its description reside in object repository. User can modify checkpoint in object repository f object changes in future. Standard checkpoint is the umbrella title for inserting checkpoint on standard objects e.g. text, image, page, link etc.

In STEP2, we are creating a page checkpoint by using active screen. Every checkpoint checks the different attribute of the object e.g. link checkpoint can checks href, innertext etc. while page checkpoint can check the number of links, number of images and load time of the page. Number of links doesn't mean that numbers need to be different to fail the checkpoint but broken links (404- page not found etc.) also can do the same though they appear to present on page.

In STEP3, we are verifying the report to know the status of checkpoint. Checkpoint sends the message to UFT report with Pass/Fail status. It does not mean that checkpoint status cannot be found during script execution. We can get checkpoint status as true/false. A single property in a checkpoint can fail the whole checkpoint (and test) so we need to carefully select the checkpoint properties criteria to check only desired properties.

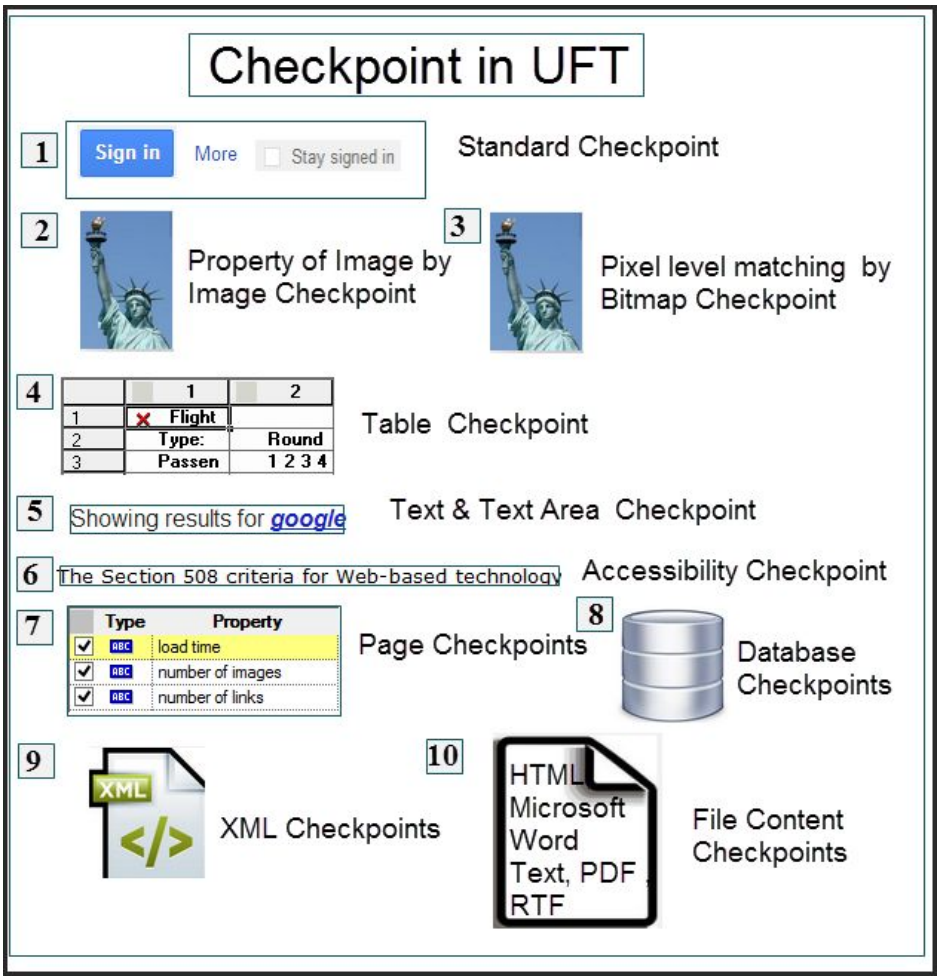
2.3 How Checkpoint Works

Checkpoint match the recorded checkpoint property with run time object property e.g. recorded page checkpoint have 60 links but during replay page have 59 links than checkpoint will fail. The Checkpoint status updated in the UFT Report.

RecordTime	Run Time	Checkpoint Status
60 Links	59 Link	
Text "Krish"	"Krishan"	
Button Enable	Button Enable	

2.4 CheckPoints Types

Checkpoints are specific of particular technology e.g. one checkpoint may supported in web application but not applicable for .Net applications and vice-versa. You can find complete list for supported checkpoint in HP UFT help file [REF]. We are discussing the commonly used checkpoint.



UFT provides the following types of checkpoints:

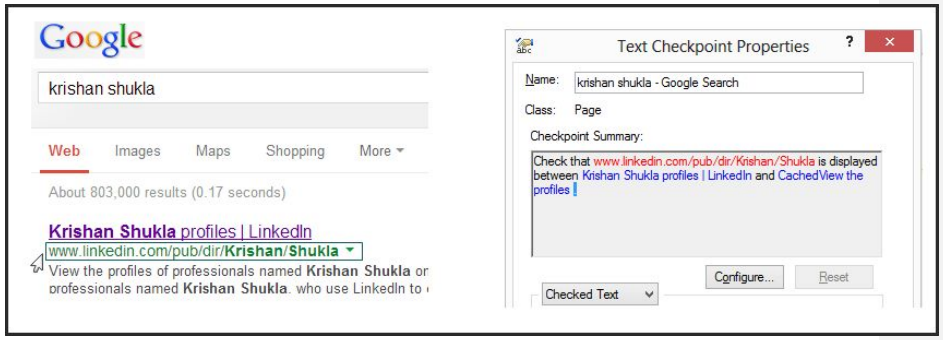
1. Standard checkpoints: Standard checkpoints are used to verify a set of properties for standard objects. This checkpoint can be used on objects like Buttons, Images, Radio buttons etc.
2. Image checkpoint: It compares the various property values of an Image like source file location, width, height etc. It does not check the pixel by pixel value of image but check the properties of the image. If looks of image changes in future but properties remain same then this checkpoint will not fail.
3. Bitmap checkpoint: Bitmap checkpoint completes Image checkpoint. Bitmap checkpoint compares an on screen bitmap images and compares it pixel by pixel against a previously recorded bitmap. We can check the area on the application or complete window itself. Also it provides the flexibility to compare the area with

finding similarity (e.g. 80% similar). Changing the computer resolution impacts this checkpoint. Bitmap checkpoint is very helpful to check the logo, banner, particular area and crucial images on the application.

4. Table checkpoint: Validate data presented in a table against predefined values. Mostly page comprises the tables for example trains timetable, price comparisons etc. Table checkpoint enables to verify the "front-end" table content on application. The tables in your application may be very large. A table checkpoint on a large table may take a long time to create and a long time to run. We can choose to include specific or all rows in table.

ConfuSense – Do not confuse with database checkpoint which checks the database table value at backend by using SQL and database connection.

5. Text checkpoint: Validates that a text presented at an expected place in the application. This is very helpful checkpoint to check the confirmation message after product sale, congratulation messages and capturing many crucial details on the page. Text checkpoints are supported only for Page, Frame, and ViewLink objects



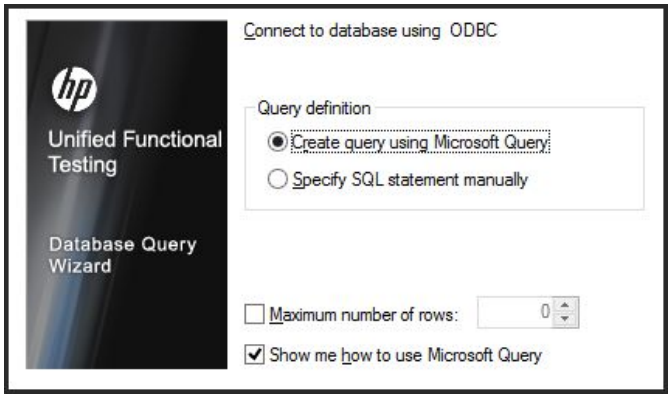
6. TextArea Checkpoint: Validate that a text string is displayed in an expected area in the application. TextArea supports multi-lines while text box allow single line content.

7. Accessibility checkpoint: Web accessibility means that people with disabilities can use the Web. People must able to interact web applications and can understand, navigate, and interact with the Web. Web accessibility also benefits others, including older people with changing abilities due to aging. To make web application W3C provide the compliance standards for web objects (like button, Image etc). UFT can check these properties by Accessibility checkpoints. [REF 1.0]. Checks for the areas in a web application to comply with W3C web content accessibility [REF] standards. The

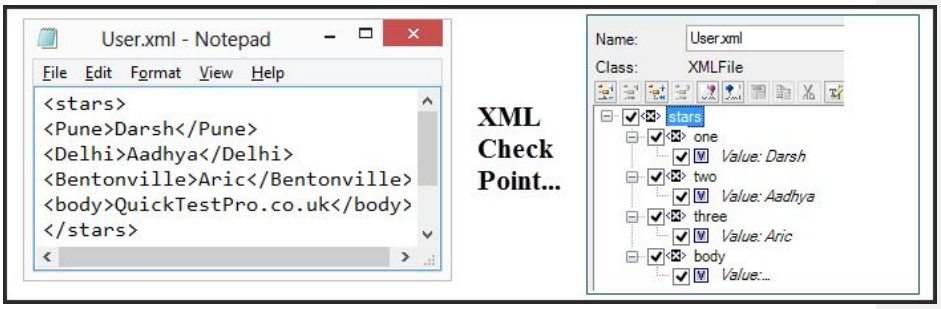
Section 508 criteria for software applications are based on access guidelines developed by the Web Accessibility Initiative of the World Wide Web Consortium (W3C). Accessibility checkpoint ensure the application fit for use by differently able people with difficulties to hear or see the application properly. Many tools [REF] can interpret/read the application object and inform user about its content and properties. So, section 508 compliant applications need to set these properties appropriately for all applicable objects. This checkpoint is very helpful for the application that requires mandatory regulatory compliance with section 508. Following are the few example of section 508 guidelines:

508 STANDARD	Pass	Fail
Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	Sufficient contrast is provided.	Contrast is poor.
Documents shall be organized so they are readable without requiring an associated style sheet.	Style sheets may be used for layout, but the document is still readable and understandable (even if less visually appealing) when the style sheet is turned off.	The document is confusing or information is missing when the style sheet is turned off.
A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	Images that have a function (images within links, image buttons, and image map areas) have alternative text which describes the associated function.	Alternative texts for linked images, image buttons, or hot spots are not descriptive of the function

8. Page checkpoint: Validate properties on a web page like numbers of links, page load time, etc. This checkpoint is very useful to check the page attribute, broken links and images on the page.
9. Database checkpoint: Validate the contents of a database entry specified in the checkpoint. To do this, we need to define a query on your database and validate the query result of query by checkpoint. Query can be define by following ways:



10. XML checkpoint: Validate the content of an XML document or a Web XML document.



XML checkpoints are not supported on Internet Explorer 9 or later running in standard mode, on Google Chrome, or on Mozilla Firefox, because the WebXML test object is not supported for these browsers.

11. File content checkpoint: File content checkpoint compares the textual content of a file generated during run-time with the source file. The source file must be located on the file system. Suppose we buy a ruby stone online and the order details generated in pdf file. UFT enables to check the content of pdf e.g. name, price etc. from the pdf file. File content checkpoint can check single, multi-line or part/whole document. This is new type of checkpoint in UFT11.0.
    Following files are supported by UFT11.52.

The Lucky Five - File Content Checkpoint				
HTML	Text	Microsoft Word	PDF	RTF

Note: XML, File content checkpoint and Database checkpoints can be added during recording or design mode (UFT | Design | Checkpoint). All other checkpoints can be added in recording mode or through active screen objects.

Exercise 6.2 of Decision based on Checkpoint status

STEP1: Create a Test to search “Krishna wikipedia”

Navigate to UFT | New | Test | GUI Test |”CheckPointLogic” | Click Create | Record | Select first radio button in web tab | http://Google.co.uk | OK | Search with “Krishna wikipedia” | Google Search button | Click Image link | Close Browser | Save | Stop recording | Save

```
CheckPointLogic*   Action1 ×   Start Page
Main
1  SystemUtil.CloseProcessByName "iexplore.exe"
2  SystemUtil.Run "iexplore.exe", "Google.co.uk"
3  Browser("Browser").Navigate "http://google.co.uk/"
4  Browser("Browser").Page("Google").WebEdit("q").Set "Krishna Wikipedia"
5  Browser("Browser").Page("Google").WebEdit("q").Submit
6  Browser("Browser").Page("Krishna Wikipedia - Google").WebButton("btnG").Click
7  Browser("Browser").Page("Krishna Wikipedia - Google").Link("Krishna - Wikipedia, the").Click
8  Browser("Browser").Page("Krishna - Wikipedia, the").Sync
9  Browser("Browser").CloseAllTabs
```

STEP2: Insert CheckPoint from Active screen

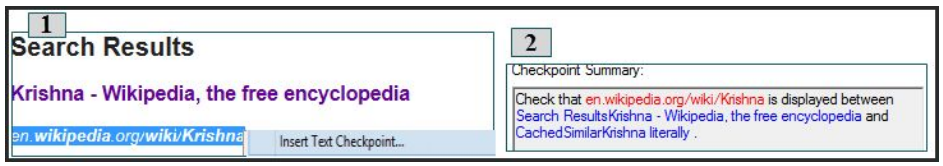
Navigate to "CheckPointLogic" test in UFT | View | Active Screen | in UFT editor, Click on line towhere UFT set "Krishna Wikipedia" in Google box | View | Active Screen | Right Click on Page | select "Insert Standard Checkpoint" | Select "Page" | OK | Uncheck load time | Click on "After Current step" | OK

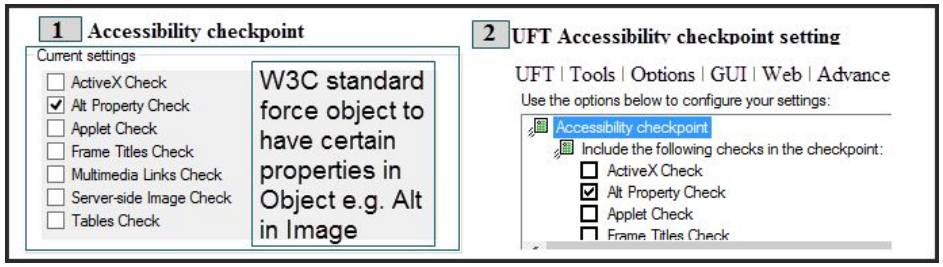
```
5  Browser("Browser").Page("Google").Check CheckPoint("Google")
```

STEP3: Insert other checkpoints

Navigate to "CheckPointLogic" test in UFT | View | Active Screen | in UFT editor, Click on 4th line to set "Krishna Wikipedia" in Google box | Click on Active Screen | Right click on Image link | | select "Insert Standard CheckPoint" | Select "Link:Images" | OK | unselect "href" | Click on "After Current step" | OK | Scroll down in Active screen | Select text "en.wikipedia.org/wiki/Krishna" | right click and Select "Insert Text Checkpoint…" | Verify the text left and right boundary | "After current step" | OK | Scroll up in active screen | Select "Accessibility Checkpoint" for Web Accessibility Testing and select Alt property | "After current step" | OK | Save | Run | View | Last run result | ALT + X + V

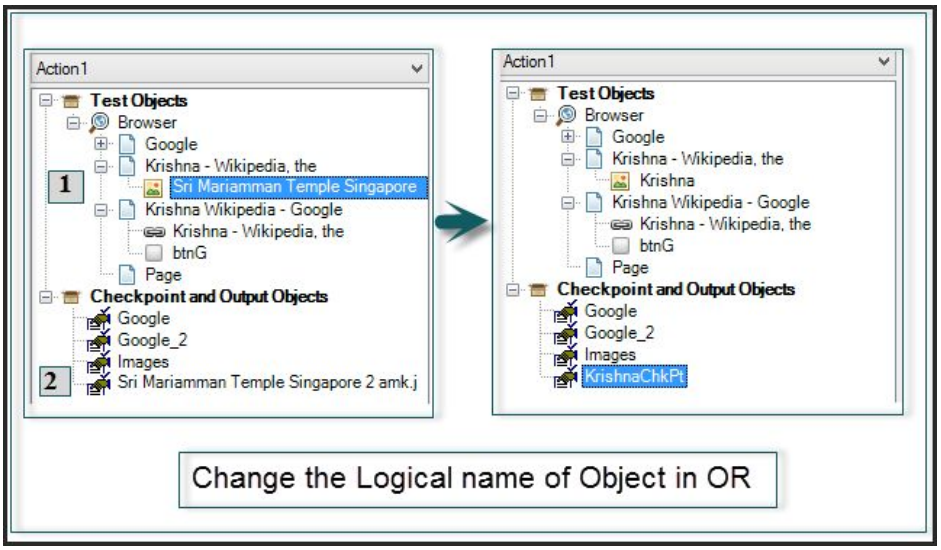
warning: Some checkpoint may fail if link and image changes in future.





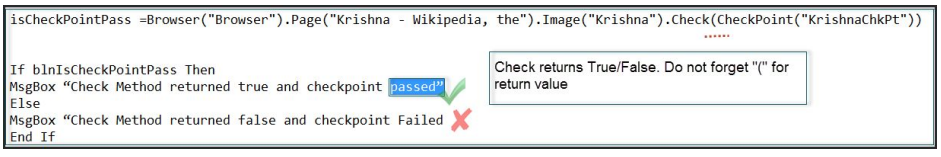
STEP4: Insert image checkpoints and modify logical name in OR

Navigate to "CheckPointLogic" test in UFT | View | Active Screen | in UFT editor, Click on line where UFT records Wikipedia page | Right Click on "Krishna" Image on Wikipedia | Select Image | OK | only select Alt, href, html tag and name properties | "Before Current Step" | OK | Save |Resources | Object repository | Navigate to Image object | Rename Image and checkpoint Object | Close | Save | Run | View | Last run result | ALT + X + V



STEP5: Programmatically decide the checkpoint return status during run-time.

Navigate to "CheckPointLogic" test in UFT | in UFT editor, Insert following line after the step "clicking wikipedia link on Google" | Save | Run | View | Last run result | ALT + X + V | Verify Checkpoints



STEP6: Modify Checkpoint properties from Object repositories

Navigate to "CheckPointLogic" test in UFT | Resources | Object Repository | Select "KrishnaChkPt" | uncheck "image type" | Set Checkpoint timeout "10" seconds | Close | Run | View | Last run result | ALT + X + V | Verify Checkpoints

2.5 Concept of Decision based on Checkpoint status

In STEP1, we are creating Google search for “Krishna wikipedia”. In STEP2, we are creating a checkpoint by active screen.

In STEP3, we are creating text, accessibility and link checkpoint. Checkpoint can be inserted before or after the current step in the script. Text checkpoint is a sandwich checkpoint. It captures the text between right hand and left hand text on the page. Text can be regularize. Suppose we are capturing a text where text contains time and date (or order number). In this case, time and date bound to change on subsequent run so eventually checkpoint will fail. We can make the regular expression of date so that UFT match the pattern of date and let any valid date occupy in the text. Regular expression are very helpful phenomena where we perform operation on string based on the pattern.

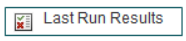
DATE PATTERN - MM/DD/YYYY or MM-DD-YYYY e.g. 09/11/1987
"(0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])[- /.](19|20)[0-9]{2}"

*There is a bug in this pattern. Find out what it is.

Accessibility checkpoint can check the ActiveX, Frame title, multimedia links, server-side images, table check and Alt property. We can configure the checkpoint at UFT | Tools | Options | GUI | Web | Advance.

In STEP4, we are creating image checkpoints that checks Alt, href, html tag and name properties. Checkpoints and output object resides in object repository. We can modify checkpoint in OR as well.

In STEP5, we modify test script flow by using the checkpoint result at run-time. Checkpoints use Check method. Any method in VBScript need to use parenthesis whenever functions needs to return the value.

Check CheckPoint …	Send result in UFT Report 
myVal = Check (CheckPoint …)	returns true or false in script 

 ***ConfuSense*** – When to use parenthesis in functions? Suppose we have myFunc function with 2 parameters. Calling myFunc a,b and myFunc(a,b) will call function during run-time but myFunc(a,b) is mandatory if it is expecting any return value from myFunc. So consider () as a gloves to receive the return value from function. This is the reason that Sub procedure never use parenthesis (gloves) to call the sub as they never expect any catching values.

In STEP6, we modified the checkpoint property. Checkpoint heart can be time bound and can be force to wait for certain duration to check the object.

2.6 One more step with Checkpoint

We have explored the most sought checkpoints e.g. image, link, text etc. but there are few checkpoint those are little difficult to understand as there are not much example available on that. We will explore File content and database checkpoint in this section.

Exercise 4.0 – Mr. UFT Clerk, File Content checkpoint

STEP1: Download London Tube map from QuickTestPro.co.uk

Open any Browser | Navigate to "QuickTestPro.co.uk" | Download London tube map from following link | Save pdf file at "C:\Test\LondonTubeMap.pdf" | close browser

http://quicktestpro.co.uk/2013/08/london-tube-map/

STEP2: Record Test on QuickTestPro.co.uk

Navigate to UFT | New | Test | "VerifyLondonTubeMap" | Create | Record | IE | Navigate to "http://Quicktestpro.co.uk/2013/08/london-tube-map/" | Click on LondonTubeMap link | Close Browser | Stop | Modify script by inserting wait (5) after opening pdf file in script | Save

STEP3: Insert File Content Checkpoint in test script

Navigate to "VerifyLondonTubeMap" test in UFT | Focus the line after "LondonTubeMap" open in UFT script | Design | Checkpoint | File Content Checkpoint | "C:\Test\LondonTubeMap.pdf" | Open | Collapse page | Check first page  *| Expand page 2 | Click on 15th lineto select "Arsenal", "Covent Garden", "Greenwich" etc. | OK | Resources | Object Repository | LondonTubeMap.pdf | Verify Object properties | Close | Save*

STEP4: Run the test to check the online tube map pdf values with source file at machine.

Navigate to "VerifyLondonTubeMap" test in UFT | Run by F5 | View | Last run result | ALT + X + V | Close

Concept of Exercise 4.0 - File Content checkpoint

In STEP1, we have created a source file downloaded from QuickTestPro.co.uk. In STEP2, we have created a test. We also synchronize the most time consuming step by inserting hard-code wait of 5 seconds.

In STEP3, we have create file checkpoint on first full page of pdf and on a single line on second page. This checkpoint is a new addition to the UFT. In STEP4, we have run the test and verify the result in UFT report.

Exercise 5.0 – Mr. UFT Banker, Database checkpoint

STEP1: Create a Database CheckPoint

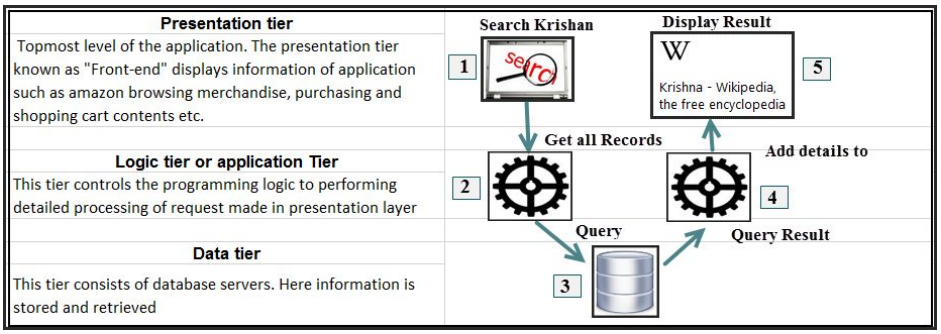
Navigate to UFT | New | Test | "DatabaseCheckpoint" | Design | Checkpoint | Database checkpoint | Next | OK | Select QT_Flight32 | OK | Select table "Flight" | Click on*  *| Next | Select "Airline", "equals" and "BA" from Column to filter | Next | Sort by "Departure" | Next | Finish*

Query:

SELECT Flights.Flight_Number, Flights.Departure_Initials, Flights.Departure, Flights.Day_Of_Week, Flights.Arrival_Initials, Flights.Arrival, Flights.Departure_Time, Flights.Arrival_Time, Flights.Airlines, Flights.Seats_Available, Flights.Ticket_Price FROM Flights Flights WHERE (Flights.Airlines='BA') ORDER BY Flights.Departure

STEP2: Record script and verify value

Concept of Exercise 5.0 - Mr. UFT Banker, Database checkpoint



Aadhya: Wow Aric! Our UFT is so smart to run the test n-number of time, parameterize by DataTable, modularize the code by reusable internal and external actions and check objects, text in files, text, database and xmls by checkpoint. That look I will never run anything manually after design my script.

Aric: Aadhya, Checkpoints are very helpful but there are few issues with them. Let me tell you.

2.7 Problem with checkpoints

UFT checkpoints are most suitable when the application being tested is static in nature like an informational website which contains static contents that rarely changes.

1. Checkpoints are not very flexible. For example, UFT does not allow the expected value of a checkpoint to be changed through the code.
2. Checkpoint cannot be created at run time (only record time or from active screen)
3. Checkpoints that contain more than 64K of data may run slowly.
4. Existing checkpoints can't copied from one Action to another action.

3.0 Verify Object by VBScript Logic – Alternate of Checkpoints

Programmatically we can check the object properties. UFT provide the following methods to get/set the Object properties (run-time or design-time)

1. GetROProperty (property of application object i.e. Google “Image” link object)
2. GetTOProperty (property of Object Repository i.e. *Link("Images") in OR*)
3. SetTOProperty
4. ~~SetROProperty~~ is not available in UFT

Exercise 6.3 programmatically object verification - an alternate of Checkpoint

STEP1: Create a test to search “krishna wikipedia” in Google

Navigate to UFT | New | Test | GUI Test |”Get_Set_TO_RO_Test” | Click Create | Record | select “Record and run on any open browser” | OK | http://Google.co.uk | Search with “krishna wikipedia” | Enter | Google Search button | Click “Krishna - Wikipedia, the free encyclopedia” link | Close Browser | Save | Stop recording

STEP2: Check the Object with GetROProperty statements

Navigate to |”Get_Set_TO_RO_Test” test in UFT | insert following lines after “Browser("Browser").Page("Krishna wikipedia - Google").WebButton("btnG").Click” line | Save | Run | View | Last Run Result | ALT + V + X | Go to Checkpoint results on the page

Wait 2

strLinkText = Browser("Browser").Page("Krishna wikipedia - Google").Link("Krishna - Wikipedia, the").GetROProperty("text")

print strLinkText

If strLinkText = "Krishna - Wikipedia, the free encyclopedia" **Then**

Reporter.ReportEvent micPass, "Wiki-Krishna Links","Link Exist on Google"
End If

```
wait 2
strLinkText = Browser("Browser").Page("Krishna wikipedia - Google_2").Link("Krishna - Wikipedia, the").GetROProperty("text")

print strLinkText

If strLinkText= "Krishna - Wikipedia, the free encyclopedia" Then
    Reporter.ReportEvent micPass, "Wiki-Krishna Link", "Link exists on the Google search"
Else
    Reporter.ReportEvent micFail, "Wiki-Krishna Link", "Link does not exists on the Google search"
End If
```

ConfuSense – We can send any message in the UFT report with status micPass, micFail, micDone and micWarning. Reporter object uses ReportEvent method to write status, Step name and Step description in UFT report

Specification of Reporter.reportevent:
Reporter.ReportEvent EventStatus, ReportStepName, Details, [ImageFilePath]

EventStatus- micPass, micFail, micDone, micWarning

Example
Browser("Browser").Page("WebPage").Image("Error").CaptureBitmap("C:\Test\logError.png")
Reporter.ReportEvent micFail, "Display Error", "Error on Page"," C:\Test\logError.png"

Concept of Exercise 6.3 programmatically object verification, an alternate of Checkpoint

We can simulate any UFT checkpoint using this technique, except a bitmap checkpoint.

Note: Equivalent Database checkpoints can be simulated using ADODB. We will find more information about ADODB in the "Working with DataBases" chapter. XML checkpoints can be simulated using the XMLUtil object.

Checkpoint check the property of object, getroproperty can also get the property and check the value

Exercise 6.4 Script to handle dynamic Objects by Set/Get RO/TO property

STEP1: Create a Test

Navigate to UFT | New | Test | GUI Test |"DymaicObjectTest" | Click Create | Record | select "Record and run on any open browser" | OK | http://Google.co.uk | Search with "krishna wikipedia" | Enter | Google Search button | Click "Krishna - Wikipedia, the free encyclopedia" link | Close Browser | Save | Stop recording

STEP2: Parameterize with DataTable

Navigate to "DymaicObjectTest" in UFT | View | Data | Double click on "A"| Provide "SearchText" Test | OK | "Gautama Buddha Wikipedia" in 1st row | Modify the script by changing line | Save | Run and let test fail | View | Last Run Result

```
'Browser("Browser").Page("Google").WebEdit("q").Set "Krishna Wikipedia"
Browser("Browser").Page("Google").WebEdit("q").set DataTable.Value("SearchText","Global")
```

STEP3: Verify the run-time object property

Open Internet Explorer |http://Google.co.uk | Search with "Gautama Buddha - Wikipedia" | Enter | Google Search button | Navigate to UFT | Tools | Object Spy | Click Spy  *| Click on link "Gautama Buddha - Wikipedia"| Link | copy text property and keep in notepad | Notice other properties of the Link Object*

Properties	Values
html tag	A
innerhtml	Gautama Buddha -
innertext	Gautama Buddha - Wikipedia, the free e
name	Gautama Buddha - Wikipedia, the free e
outerhtml	<a href="http://en.wikipedia.org/wiki/G
outertext	Gautama Buddha - Wikipedia, the free e
target	
text	Gautama Buddha - Wikipedia, the free e

STEP4: Modify the script and objects at run-time

Navigate to "DymaicObjectTest" in UFT | Insert code script after "Browser("Browser").Page("Krishna wikipedia - Google").WebButton("btnG").Click" line by providing text property in SetTOProperty statement | Save | Run | View | Last Run Result | ALT + V + X

```
Select Case DataTable.Value("SearchText","Global")
Case "Gautama Buddha Wikipedia"
Browser("Browser").Page("Krishna Wikipedia - Google_2").Link("Krishna - Wikipedia, the").SetTOProperty "text","Gautama Buddha - Wikipedia, the free encyclopedia"

End Select
```

```
 9  Select Case DataTable.Value("SearchText","Global")
10  Case "Gautama Buddha Wikipedia"
11  Browser("Browser"). _
12  Page("Krishna Wikipedia - Google_2"). _
13  Link("Krishna - Wikipedia, the"). _
14  SetTOProperty "text","Gautama Buddha - Wikipedia, the free encyclopedia"
15  End Select
```

 ***ConfuSense*** – The line continuation '_' (underscore) character permits a statement to span multiple lines. So the line 11 to 14 is one single statement.

STEP5: Modify the script and objects at run-time

Navigate to "DymaicObjectTest" in UFT | View | Data | Provide "Krishna Wikipedia" in 2nd row | Save | Modify script in "Select case" as given below | File | Setting | Run | "Run on all row" | OK | Save | Run | View | Last Run Result | ALT + V + X

```
Select Case DataTable.Value("SearchText","Global")

Case "Gautama Buddha Wikipedia"
Browser("Browser").Page("Krishna Wikipedia - Google_2").Link("Krishna - Wikipedia, the").SetTOProperty "text","Gautama

Case "Krishna Wikipedia"
Browser("Browser"). _
Page("Krishna Wikipedia - Google_2").Link("Krishna - Wikipedia, the"). _
SetTOProperty "text","Krishna - Wikipedia, the free encyclopedia"
End Select
```

4.0 Back on Checkpoint - Checkpoint Properties

UFT does not allow modifying checkpoint properties at run-time though it is possible to configure a checkpoint to use a DataTable value or an Environment variables.

We can Set/Get the checkpoint property by

1. SetProperty (notice it is not SetTOProperty that used to set Object property in OR)
2. GetTOProperty

SetProperty will change the checkpoint property and GetTOProperty will return the property value in OR.

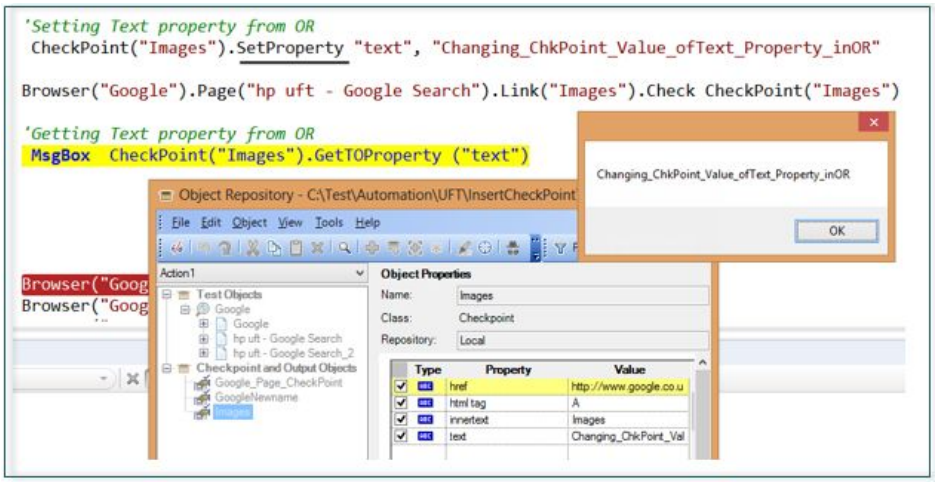
Exercise 6.5 Checkpoint modification during run-time by Set and Get property

Step1: Create Test

Navigate to UFT | New | Test | GUI Test |"ModifyCheckPointTest" | Click Create | Record | Select second radio button in web | http://Google.com | OK | Search with "HP UFT" | Enter | Search button | click Standard Checkpoint | Click Images Link | Select "Link: Image" as in Figure | uncheck href | keep other properties | OK | Click on Images Link | Click on Search Link | Close Browser | Stop recording | Save

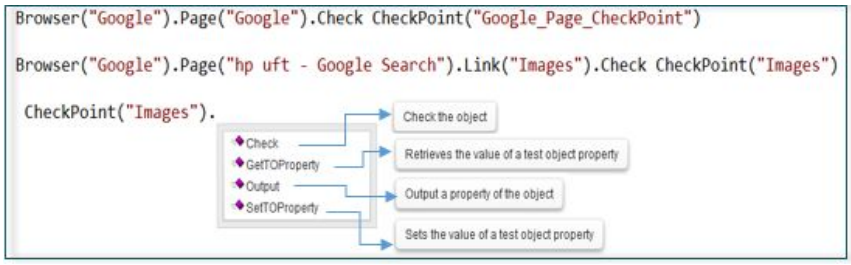
Step2: Modify checkpoint in object repository

Navigate to" ModifyCheckPointTest" test in UFT | Modify script | Test | GUI Test |

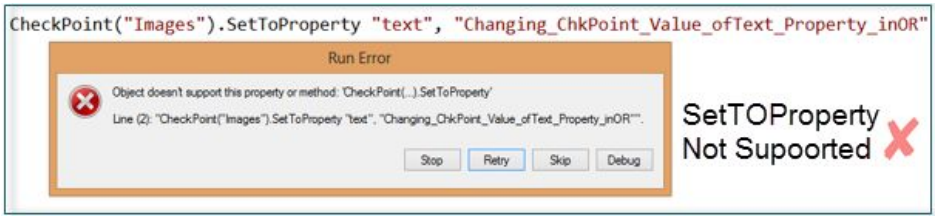


Concept of Exercise 6.5 Checkpoint modification during run-time by Set and Get property

Checkpoint Support the following method in UFT.



UFT has a bug. UFT IDE shows SetTOProperty for checkpoint but setting the property using it will gives following error. Property ”SetProperty” is undocumented and should use with caution.



Warning: It is not recommended to change the checkpoint properties using the SetProperty method as this is undocumented method and might not work in later version of UFT. Also HP may not provide support for the error by this method in future.

Aadhya: Aric! Checkpoint, what a lovely feature! I will check everything now. No manual checking for links, images, buttons, text, text area, xmls, database, screen tables, page, bitmap and accessibility for any object now. My brother is very naughty, can UFT check his mind as well?

Aric: Sorry Aadhya, you forgot. UFT only recognize the object with associated Add-ins. There are three free add-ins web, Visual basic and ActiveX. Humans are still unrecognized by UFT.

Aadhya: Aric, in our company, we want to run test from various machines with different operating system. User name will also change for every machine. I need our automation to have all the information of test, user, operating system and action iteration information. Also I saw that in DataTable use for parameterization but I need more simple way to get data. I want to provide name and get value back. Let me explain, we have country, capital name pair than if I provide country name function should return capital name.

Aric: Ok! Let's play a game. I will say a word and you need to reply with a word that comes first in your mind. OK, France
Aadhya: Paris
Aric: UK
Aadhya: London
Aric: USA
Aadhya: Washington DC
Aric: UFTSuperStar
Aadhya: Aadhyaaaaa!
Aric: ☺

In same way UFT provide environment variable. You ask UFT a variable name and UFT return the value of it.

Environment. Value ("UFTSuperStar") should return "Aadhyaaaaa!"

Aadhya: oh! UFT, the GOD knows everything?How it knows UFTSuperStar name have value Aadhyaaaa! ?

Aric: No, No, Aadhya UFT is not God and we need to set these values before our test run. UFT just keep them available during the whole test (in all actions in test).

5.0 Environment variable

UFT environment variable are variables whose values persist across and are shared by all actions in test during the test execution.

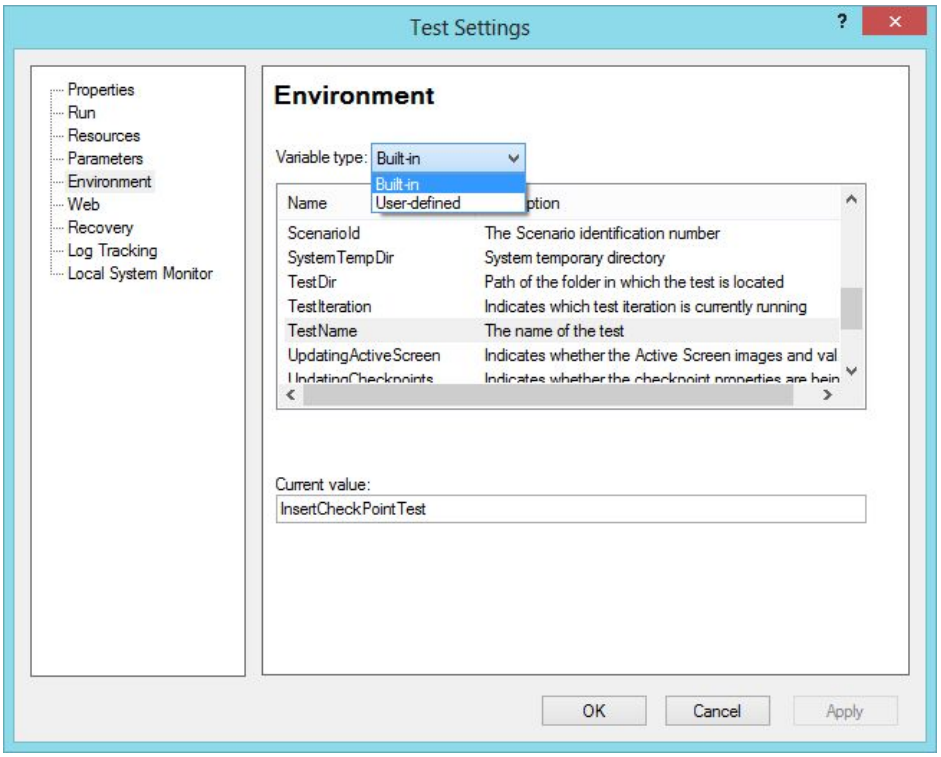
A Glance on Environment variable

Exercise 6.5 Understanding Environment Variable

Step1: Open environment variable settings and Verify Built-In environment variables

Navigate to UFT | Open insertCheckPointTest | File |Settings… | Environment

Navigate to UFT | New | Test | GUI Test | EnvVariableTest | Click Create | Record | Select "Run and record on any browser" | OK | http://Google.com | OK | Search with "London Wikipedia" | Enter | Search button | Click on "London - Wikipedia, the free encyclopedia" | Close Internet Explorer | Save | File | Settings… | Environment

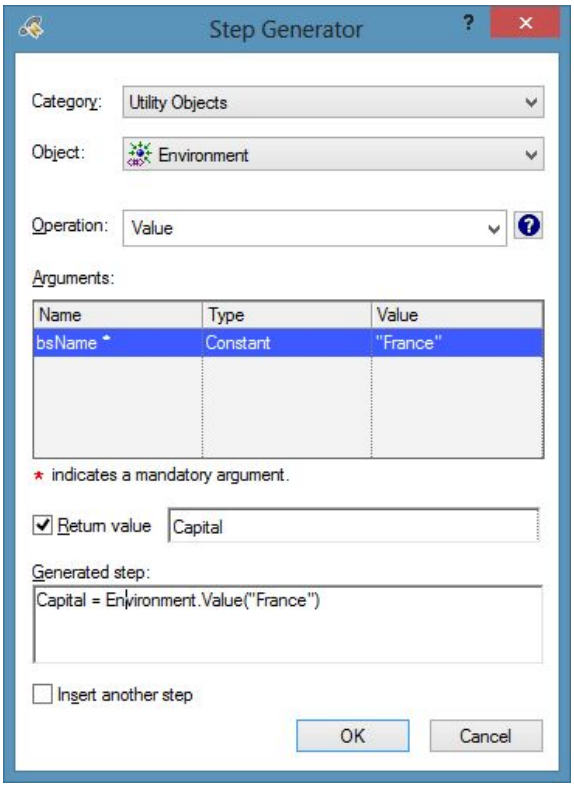


Step2: Set user defined environment variable

Navigate to UFT | Open EnvVariableTest | File |Settings… | Environment | Select variable type "User-defined" | Click "+" | Provide "France" and "Paris" in name, value | OK |Apply | OK | Save | Close UFT

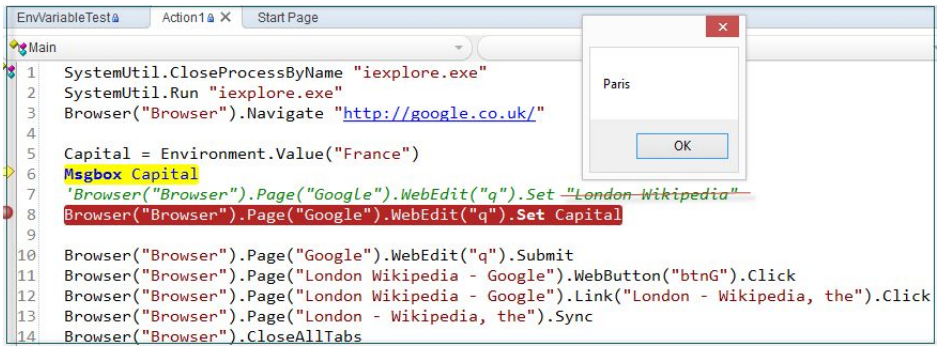
Step3: Get the value of Environment variable in Test Script

Navigate to UFT | Open EnvVariableTest | Action1 | Right Click on the beginning of the first line | Insert Step | Step Generator | Select Utility Object in category | Select "Environment" in Object | Click Value column in first row in Argument table | Select "France" | Provide "Capital" in Return Value | Press Tab key | Verify Generated Step "CapitalName = Environment.Value("France")" | OK



Step4: Run the Test Script and search with Paris

Navigate to "EnvVariableTest" test in UFT | Modify the script



Concept of Exercise 6.5 Understanding Environment Variable

Environment variable Types and Using in Script

`There are two types of Environment variables in UFT

1. Built-In (Read-Only) – UFT provides the information related to Test, action, system operating system and its version, computer name (local host name). These property are read-only and cannot be modified.
2. User-Defined
   2.1 Internal (Editable) – These variables are defined in a test and saved with the test. These variable can be modified during run-time e.g. France in our previous exercise.
   2.2 External (Read-Only) – These variables are defined in an external environment file. These variables are read-only. The external file can be associated with the test and can also be loaded at run-time using the LoadFromFile method.

Commented [ks2]: 2nd Edition

We have seen Built-In and Internal user defined variable in UFT. External User-Defined, defined in external file and associate with test either by UFT GUI or programmatically.

Exercise 6.6 Understanding External User-Defined Environment Variable

Step1: Save the following text in myExternalEnvVariable.xml in your system (e.g. in C:\Test)

```
<Environment>
        <Variable>
            <Name>France</Name>
            <Value>Paris</Value>
        </Variable>
        <Variable>
            <Name>India</Name>
            <Value>Delhi</Value>
        </Variable>
        <Variable>
            <Name>UFTSuperStar</Name>
            <Value>Nena</Value>
        </Variable>
</Environment>
```

1 2 3

Step2: External User-defined variable in Test Script by UFT Test Settings

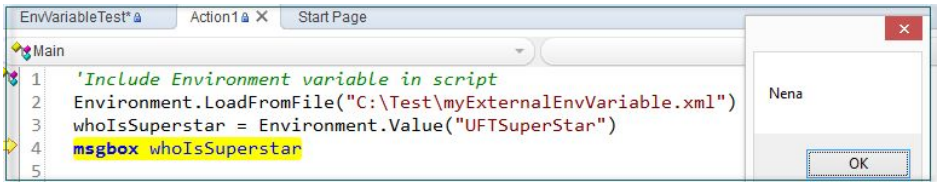
Navigate to UFT | Open EnvVariableTest | File |Settings… | Environment | Select variable type "User-defined" | Check "Load variable and values from external file" | Provide xml file path | No | Verify the variables | Apply | OK | Save | insert "whoIsSuperstar = Environment.Value("UFTSuperStar")" at second line | Save

Step 3: External User-defined variable in Test Script by programming

Navigate to UFT | Open EnvVariableTest | File |Settings… | Environment | Select variable type "User-defined" | Uncheck "Load variable and values from external file" | Apply | Save

Step4: Associate file with Environment.LoadFromFile at run time.

Navigate to "EnvVariableTest" test in UFT | insert following lines



5.1 Advantages of Environment variable

1. A way of Parameterization in UFT

2. Global variable and available for whole test

3. Program to make data/variable read-only or editable

4. Provides critical information e.g. system OS information, test, action and user property during run-time with minimum effort.

Aadhya: I am confuse with the name, environment, why it is so?

Aric: Environment variables are strings that contain information such as drive, path, or file name. UFT is one step ahead and also let user to keep their variables. Environment variables is universal concept and every computer have environment variables.

 Navigate to Control Panel | System |Advance System Settings | Advance | Environment variable

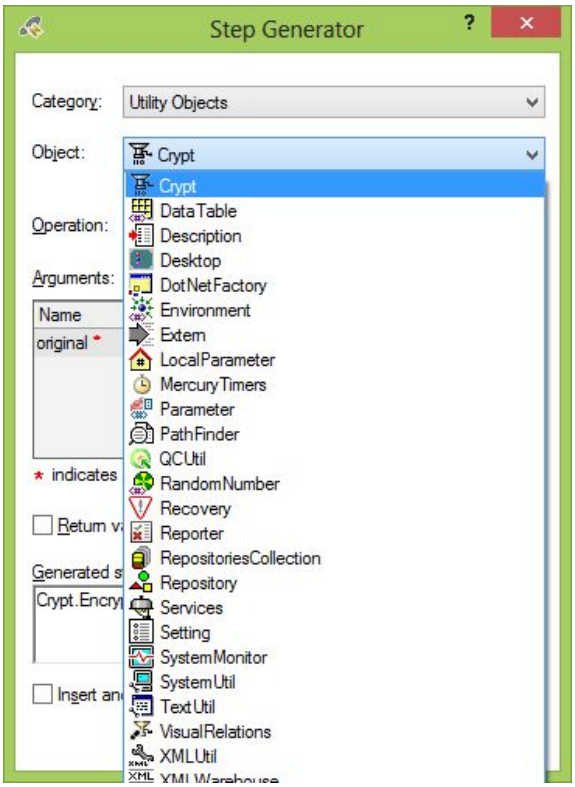
Summary:

Aadhya: what else UFT can do? I am dying to know. Please just tell me so I can go back in my company and can show the magic of automation.

Aric: Ok! UFT is very powerful tool for automation because it provides lots of helpful utilities.

6.0 Utilities Statements

 *Navigate to UFT | New | Test | Select GUITest as "UtilityTest"| Create | Second radio button on Run and Record Settings | OK | Search with "HP UFT" | Enter | Click on "Image" Link | Click on "Search" Link | Close Browser | Stop recording | Save | Go to Action1 |Right Click on the beginning of the first line | Insert Step | Step Generator | Select Utility Object in category | Click on Object*



Reference:

REF : Supported Checkpoints for each UFT Add-In in UFT Help file

UFT User Guide > Appendix > GUI Checkpoints and Output Values Per Add-in > Supported Checkpoints

REF1.0 http://www.w3.org/WAI/intro/accessibility.php

REF Accessibility Checkpoint http://www.w3.org/standards/webdesign/accessibility

REF Web Accessibility Evaluation Tools http://www.w3.org/WAI/RC/tools/complete