Chapter -3

UFT – FIRST STEP

“Take the first step in faith. You don't have to see the whole staircase, just take the first step.” - Martin Luther King, Jr. quotes

Let’s explore the UFT world. UFT can record the user steps and mimic them during replay (Record once run many times philosophy).

In this chapter you will learn following topics <Revise: Create a Image>

1. Record First Test in UFT
2. Understanding the Application Object and UFT
3. Object Identification Process by UFT
4. VBScript and UFT
5. Understanding UFT IDE
6. Enhance Test Script to interact at run-time
7. Solution in UFT
8. Shared Vs. Local Object Repository in UFT

1.0 Record First Test in UFT

In this chapter we will learn UFT basics and turn ourselves in UFT automation star. Aadhya, our future automation superstar, is also looking ahead to learn and earn the automation star status. Let's see what she is discussing with our mentor Aric.

Aadhya: Hi Aric, I have installed UFT on my machine. You told me automation phases and the scenarios to do automation. But, I want to learn UFT. I have little knowledge of programming language.

Aric: Sure! UFT provides very smooth learning curve. So you need not to worry about it. Let's record the first test.

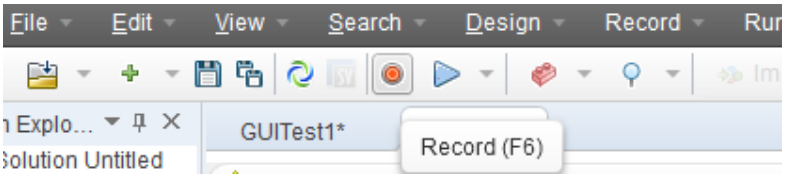
START- Exercise – 1.0 (of 4-Steps): Record and Run first test in UFT

STEP1: Create New Test

Open UFT | New | GUI Test with Name 'GUITest1' | Create

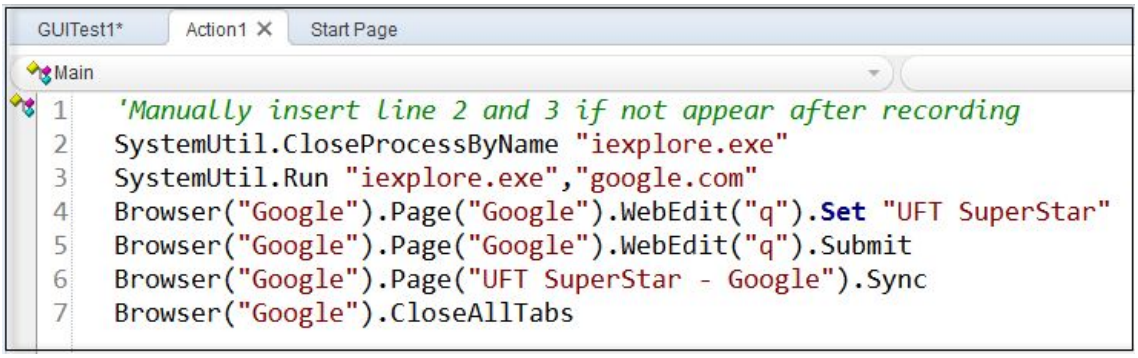
STEP2: Start recording mode in UFT

Click Record button (or F6 key) | Click 'Ok' with default record and run settings with first radio button



STEP3: Record Test Steps to search "UFT Superstar" in Google.

During recording, open Internet explorer | Google.com | Type "UFT Superstar" in Edit Box | Enter | Close Browser | Stop recording by clicking stop | Save As "FirstTest" | Modify Test



STEP4: Replay Test Steps

Navigate to "FirstTest" in UFT created in Step-3 | Click Run (or press F5 key) on UFT | Go to View | Last Run Results | ALT + V + X |Verify results | Save

Concept of Exercise - Understand Record and Replay mechanism in UFT

In STEP1, we are creating a GUI Test. User can create different type of test, for example, API, API Load, Business Process Test etc. In STEP2, we are starting recording on any open browser. We can also choose a specific URL to start our recording. In STEP3, we are recording the sequence of our test case. This recording phase is also known as design-time phase. Finally, In Step4, UFT replaying the test (known as run time phase).

UFT mimics the user actions and record the steps in Actions. During recording, UFT interacts with application's objects (e.g. buttons, menu, links, labels, tables. edit boxes, drop downs etc.) and record the operation perform on those objects.

Let's explore the line

To reach at Google WebEdit box, UFT identifies Browser object and name it "Google" so Browser ("Google") is the top level "parent" object. All the objects on Google page reside under Google Page object. Remember the first "Google" object (i.e. browser object) is different from the second "Google" object (i.e. page object). Finally we reach at the WebEdit object on which we need to perform action. UFT records object properties and gives logical name "q" during recording. So Object "q" represents Google editbox in UFT. UFT provides set of operations* on every object. In this case UFT use Set method to set the string "UFT SuperStar".


ConfuSense – Operation are supported actions on particular object e.g. click, set, select, Submit etc.

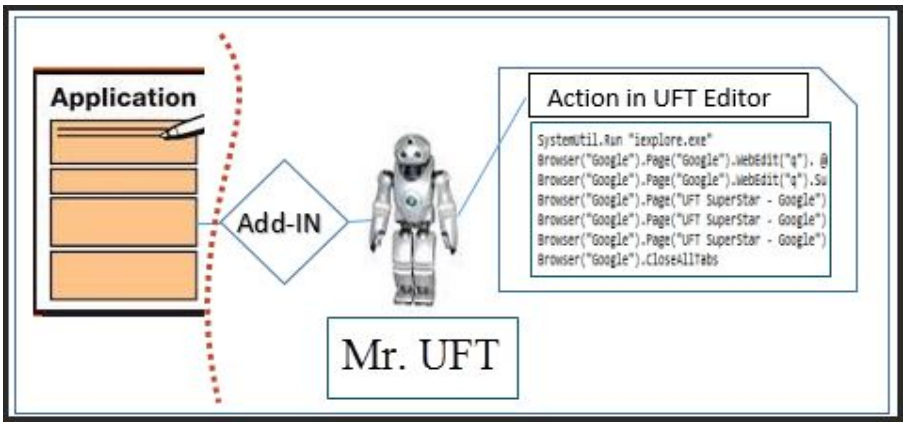
*ExplorySTEP1: Explore all the method supported by the object in UFT

Navigate to "First Test" in UFT | replace ".Set "UFT SuperStar" with "."

UFT provides IntelliSense to show all the method and property supported by the object.

UFT recognizes the application if it has been associated with correct Add-In. Add-in helps UFT to recognize the application object during recording.

All recorded steps records in action script in UFT. Each UFT test comprises **actions**. When we create a new test, it contains a call to single action (i.e. Action 1). UFT Script can be enhanced by introducing the VBScript logic or QTP in-built utilities in script.



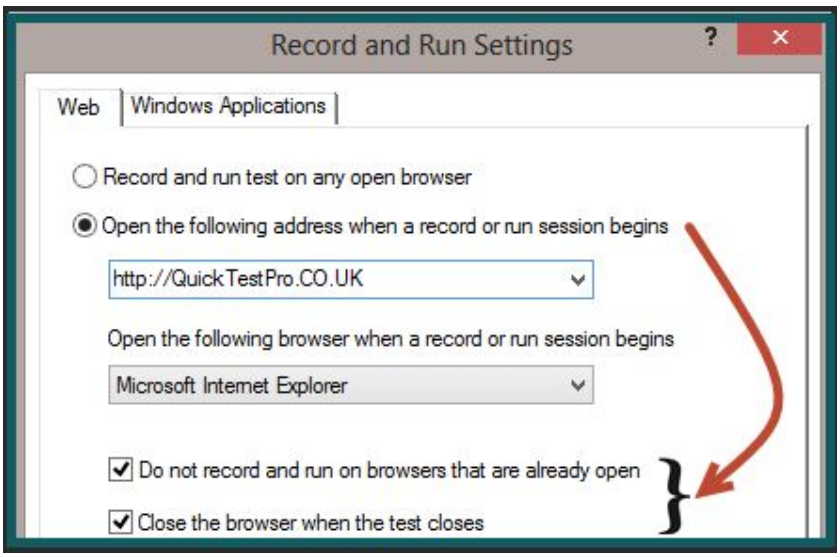
In Step2, before recording start-up, there was a Record and Run setting wizard for run and record settings. This wizard let us to configure recording on pre-define application or allows us on any application.

UFT provide run and record options for Web and Window applications.

A. In web "Run and Record setting" , user can start recording on any URL by explicitly opening the browser by selecting first radio button.
   If user wants UFT to open the browser and navigate to particular URL then second radio button should be selected. In this option we can also chose to close browser automatically after finishing the recording.

Explore STEP1: Open the run and record setting in UFT

Navigate to "First Test" in UFT | Select "Record" in UFT Menu | Record and Run Settings… | Web

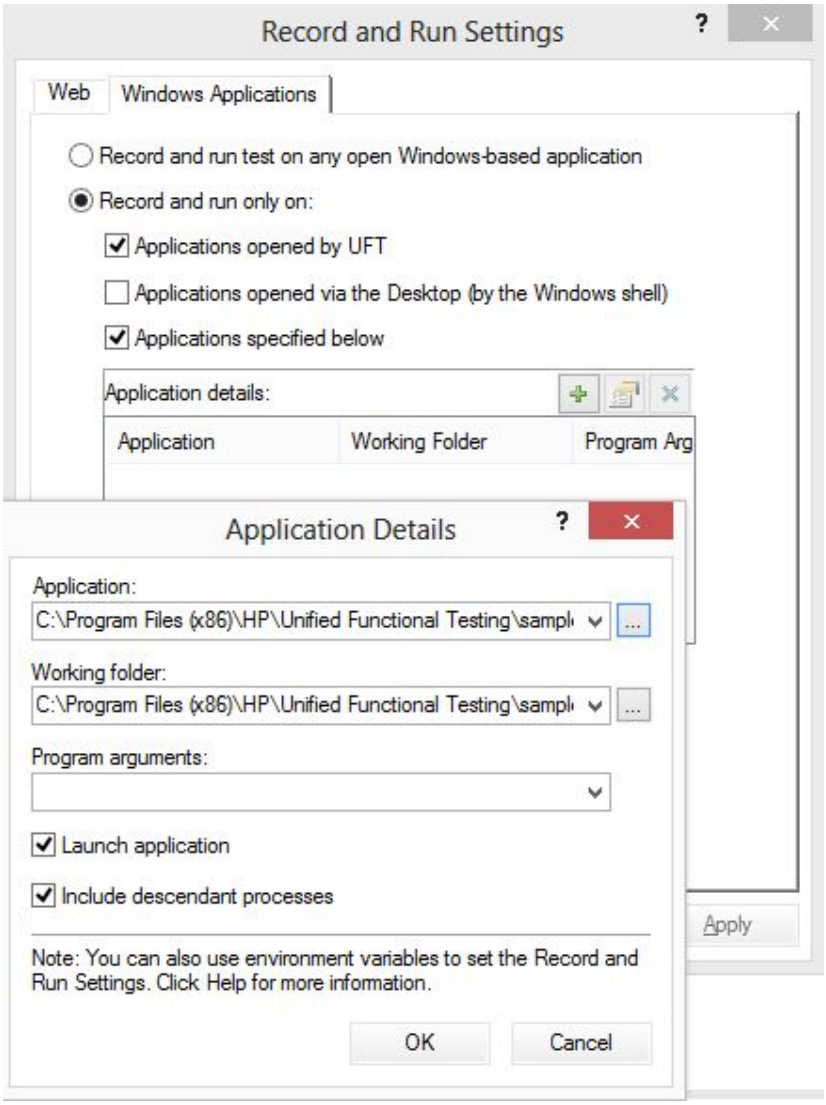


B. In Windows applications' "Record and Run Settings", user can start recording on any window application (e.g. Notepad, calculator etc) by selecting first radio button. If user wants UFT to

record on specific window application then (s)he need to select the second radio button. In most cases, we specified the .exe location of application.

Explore STEP2: Open the Windows run and record setting in UFT

Navigate to "FirstTest" in UFT | Record in UFT Menu| Run and Record Settings… | Window



Key Learning of Exercise: UFT can record user actions with help of suitable add-in. These steps goes into action script. Application built in various technologies can be access (and record) by either windows or web GUI interface so UFT provides record and run setting for that. UFT mimic same user steps recorded in script during replay.

END- Exercise – 1.0 (of 4-Steps): Record and Run first test in UFT

Aadhya: Wow! I ran these steps and automated Google search. Can I run it only on web applications?

Aric: Aadhya, Well, you can run test on applications build on various technologies* e.g. windows desktop (like notepad, calculator), Java, .Net, Oracle, SAP, Siebel and many more by buying suitable add-in for that technology. By default (and with no extra cost) UFT supports Active X, Visual Basic and Web Add-in. So, Aadhya you should also try recording on windows applications for example Notepad and calculator and see how it works.

Aadhya: Thanks Aric! I will. What is this magic, how UFT is able to mimic same steps again? What is this recording language, I think it is not VBScript. Can you please tell me how UFT works?

Aric: Yes, we will discuss following object internals

2.0 Understanding the Application Object and UFT

Application is a summation of different type of objects. Learning the chemistry between UFT and these objects are crucial. Let's explore the following topics.

1. What is Application Object
2. Object Spy to verify Object Properties
3. Object storage in UFT – Object Repository
4. Relation between Script and Objects


FindingAadhya: You can verify technologies supported by UFT @ UFT | Help | What's New Help > New Supported Operating Systems and Environments > New Supported Environments for GUI Testing

When we record UFT test, broadly following steps happen automatically in UFT

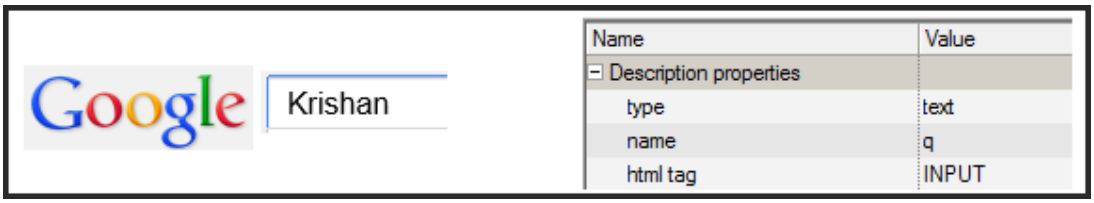
1. UFT **records** the script that is equivalent to user actions in Action 1
2. UFT **stores** object in Object Repository.

The most important thing in automation is to identify the object and perform user action on that object. For example during recording if we click on any button, UFT stores button identification properties in Object repository and simultaneously write step in Action 1 Script. So first we need to explore what is Object and UFT-Object relation.

2.1What is Application Object

London may be your ideal location to live, but what comes in mind by this name? May be many attributes e.g. famous monument, Tower Bridge, Themes River, London Eye, Westminster, London Stock Exchange etc. So every physical substance can be identified by unique attribute. Same is true for application object. The Object comprises the collection of attribute (properties). Combining these properties makes every object identifiable on application.

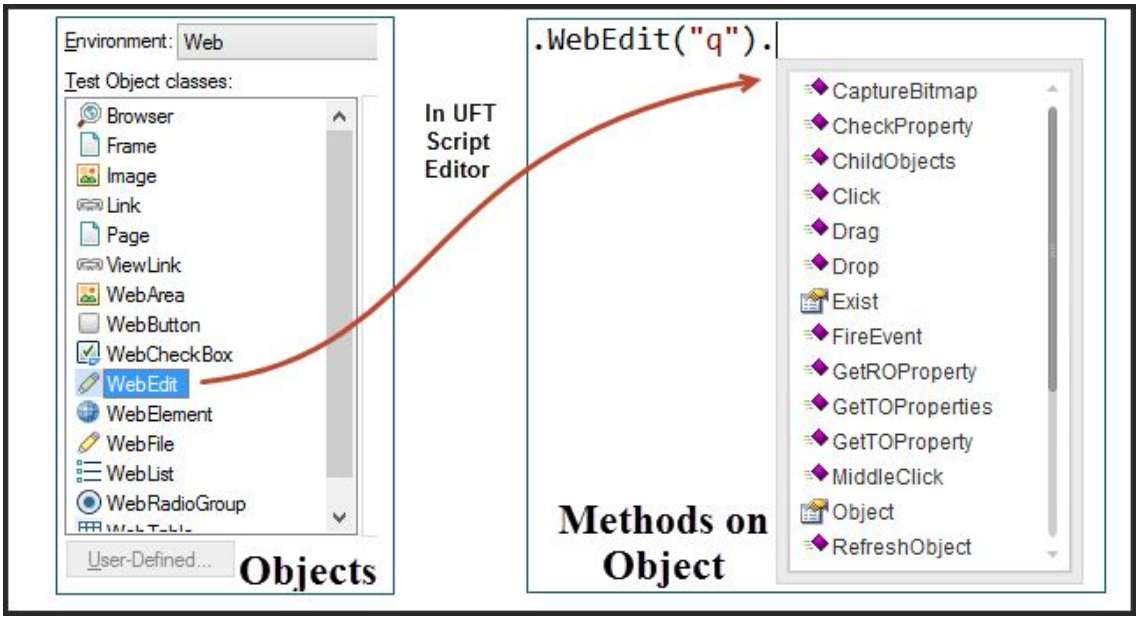
Object is a collection of properties e.g. WebEdit can have Type, name and html tag properties.



Add-in enables UFT to recognize the application objects (e.g. for web application - Browser, Page, Button etc.) based on their unique properties set. We can perform specific action on particular object e.g. string can be set in editbox and navigation on particular URL can be perform on Browser object.

STEP1: Explore the Web objects supported by UFT

Navigate to "FirstTest" in UFT | "Tools" | Object Identification… | Select "web" | WebEdit | Close | Go to UFT Action 1 | verify all method supported by WebEdit in Script.



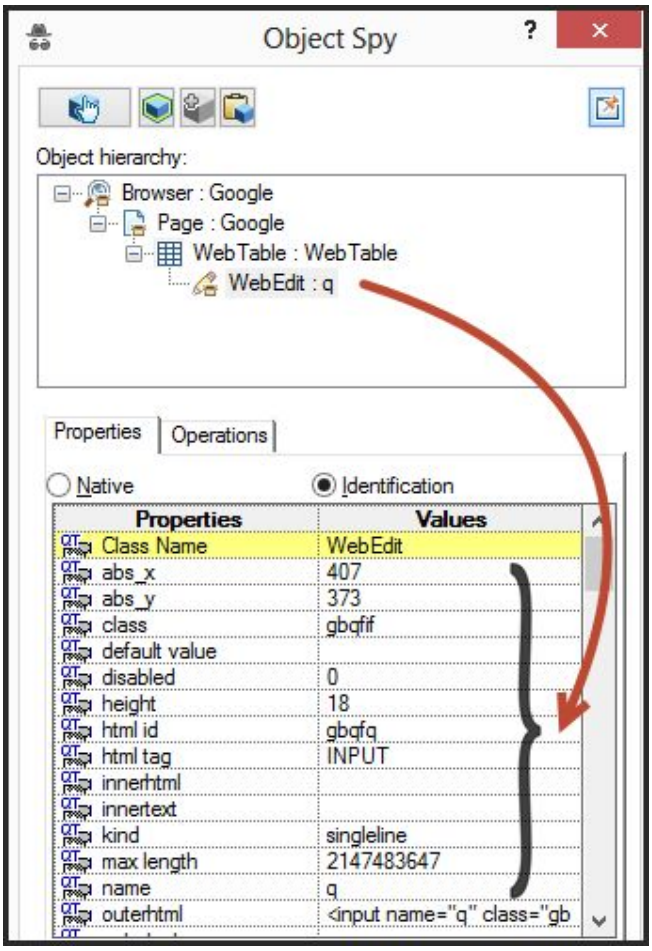
Every object have different properties-methods set. The value of property is given during development of application. Most of the time developer provides minimal set of property-value to create the object on application. These property sets captured by the UFT with the help of Add-In and stores in Object Repository.

2.2 Lets Spy the Object Properties

The Object spy is used to interrogate methods and properties* supported by an object. It displays all the available properties in the Properties tab and all available methods in the Operations tab. To understand the application objects, we can spy them to learn the supported properties & methods.

STEP1: Men in Black – Spy the object

Open Google.com in IE | Navigate to UFT |File | New | Test | Create | Tools | Object Spy | click Spy Icon  *| Click on Google Search Box*

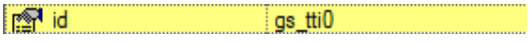


ConfuSense – Object is a collection of properties. Properties can be anything e.g. height, text, inner-text, html id etc. The collection of these properties dictate the behaviour of that Object in application.

iNTERVIEW on Object Spy

Que1: What is object Spy and what information does it provide?
Ans: Object spy is UFT tool which helps us spy on objects in the Application and see their methods and properties. It has two radio buttons: one for properties (Native/Identification) and the other for the methods supported by the different types of objects.

Que2: What are Native and Identification properties in Object Spy?
Ans: Identification properties are the default property set used by UFT to recognize the specific object. Native properties are specific to application and can be customize by development. For example if development team decide to provide the unique id to each object that UFT doesn't capture as a standard property then it can be intrigue by spy and used for identification during run-time. These property can be access by .Object during run-time or they can be used in descriptive programming.



Que3: Can you add object in OR by using spy tool, /highlight object or copy the properties?
Ans: Yes. We can add it in Object Repository and copy the properties (known as copy on clipboard).

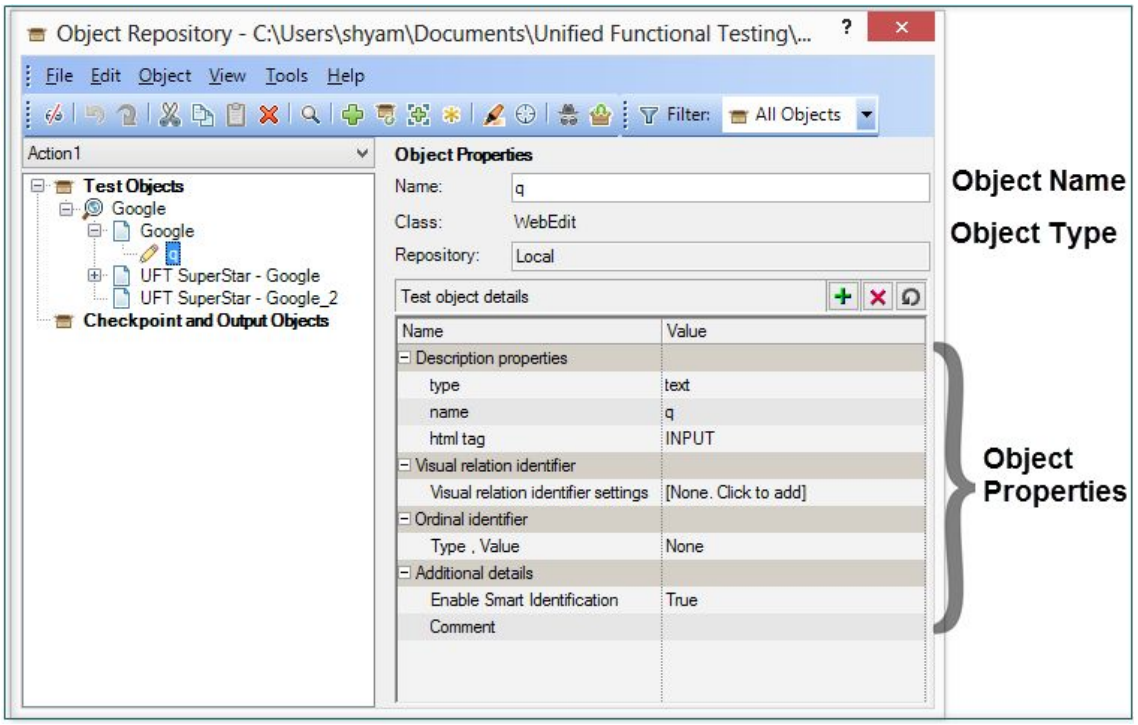
Que4: if you click spy it give no oppurtunity to navigate between screens. Have you faced it?
Ans: Yes, Pressing Control (CTL) key during spy will enable navigation between applications.

2.3 Where Objects store in UFT – Object Repository

UFT stores properties of each object in Object Repository (OR) during recording. These object called as “Test Objects”.

STEP1: Verify Objects in Object Repository

Navigate to “FirstTest” UFT | Resources | Object Repository



During the test run UFT match the properties of an Object in OR (Test Object) with real Object in application (Run time Object). UFT performs the operation written in the action script for the matched object.). If UFT unbaled to find the object during run-time similar to OR properties in application then it flag the error.

Suppose if property of Object changes in application then UFT will unable to recognize object at run time. In this case, we should add or modify correct properties in OR.

1	During Recoding	Object stored in Object Repository (known as TestObjects)
2	During Replay	UFT picks object property from repository and match with object in current application (RunTimeObject)
	TestObject == RunTimeObject 	

We can highlight the object in application to verify that test object match with runtime object.

Navigate to Internet Explorer | Google.com | Navigate to "FirstTest" in UFT | Resources | Object Repository… | Go to Object "q" | Click on Highlight button

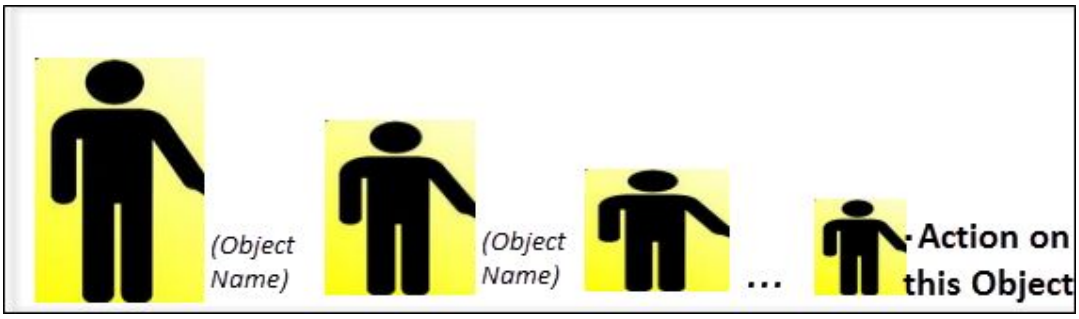
1	Add Object to the repository
2	Update the modified object
3	Add object based on the Insight (image) identification
4	Define property set when object is not available
5	Highlight the object in application
6	Spy the Object
7	Associate object repository of other tests

2.4 Relation between Script and Objects in Object Repository

If you have noticed the hierarchy in test script and Object repository you may find the relation between them.

```
Browser("Browser").Navigate "https://www.google.com/"
Browser("Browser").Page("Google").WebEdit("q").Set "UFTSuperStar"
Browser("Browser").Page("Google").WebEdit("q").Submit
```

These steps can be correlated with OR hierarchy and can be understood as following:
GreatGreatParent (ObjectName)…Parent(ObjectName).ObjectClass(TestObjectName).Operation



So When we perform any operation on application, UFT captures Object in OR and record steps in editor with same hierarchy as in OR.

3.0 Object Identification Process by UFT

Do you recognize the following picture? How your mind process to identify this image (object)? We combine many properties to uniquely identify this bridge e.g. Type of bridge, title, surroundings etc.

There is 3-If concept to recognize this bridge, first we search most important properties that can identify object uniquely e.g. name, title, banner etc. But if (1) we do not find the description then we try to add more "assistive" description like size, colour and material. If (2) still we are unable to recognize than we try to identify this from the "location" (or order of occurrence). If (3) still we are unable to identify it, then we try our "smartness" and solve the description by identifying basic properties by filtering out other objects that do not fit with it.

UFT works in same way. UFT learn Object property during record time and try to match them during run time. These are following properties set UFT uses to identify any object.

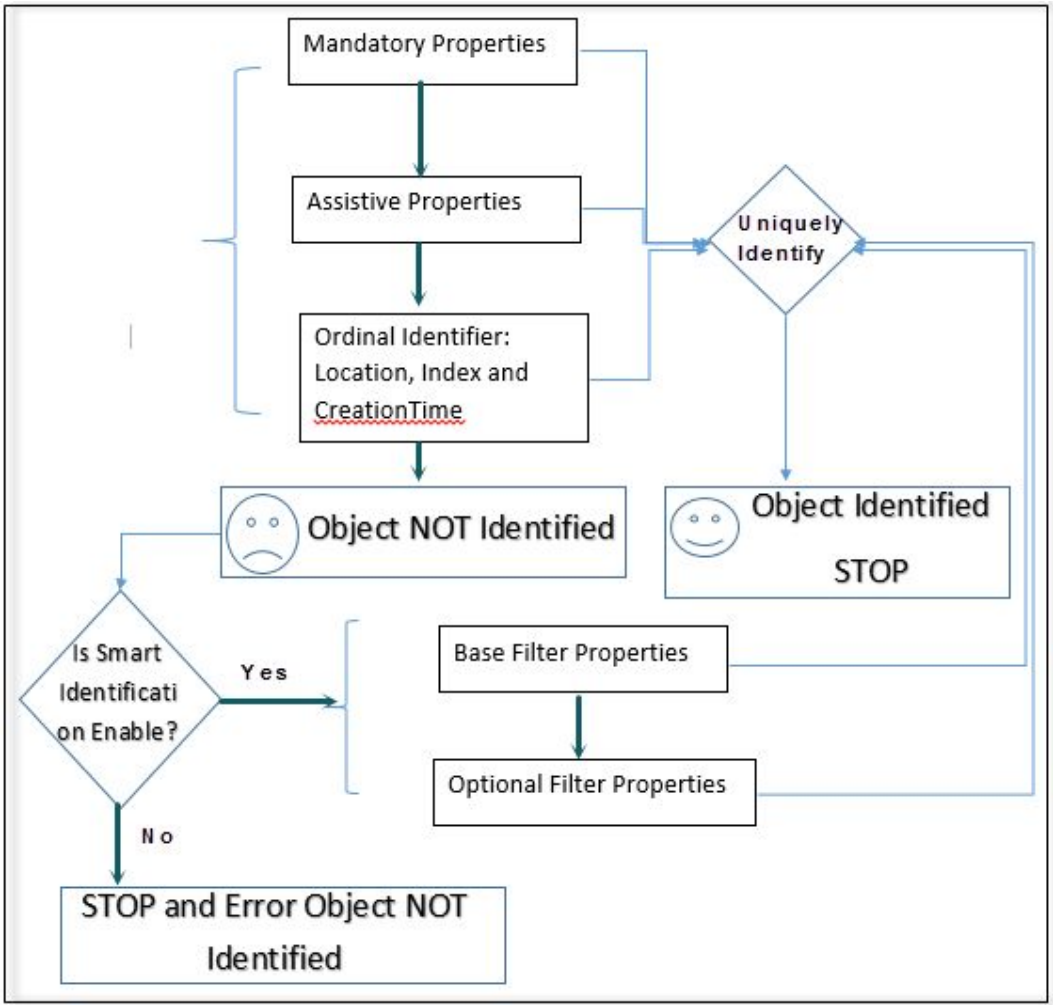
1. Mandatory Properties
2. Assistive Properties

3. Ordinal Identifier

If Smart Identification is Enable – UFT restart identification by forgetting 1-3 properties

1. Base Filter Property
2. Optional Filter Properties.

UFT have the following order to recognize the Object.



Aadhya: Why we need so many properties? Who will decide which properties are mandatory and which are assistive?

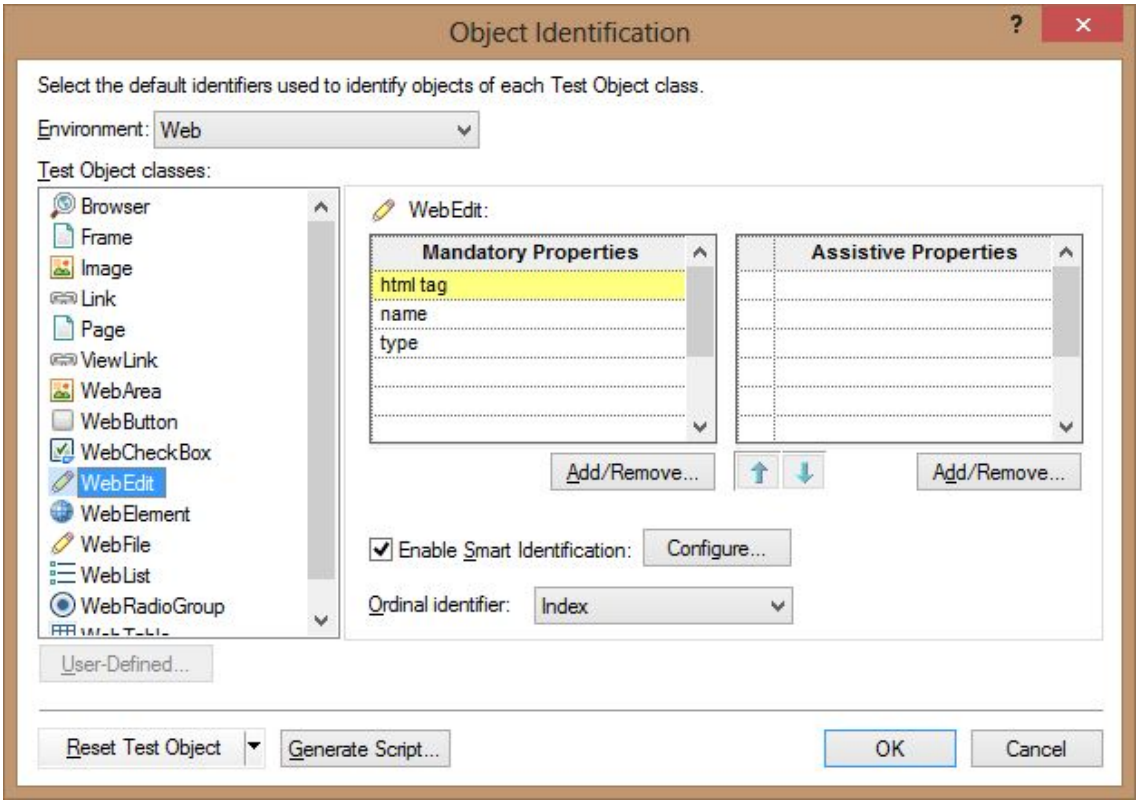
Aric: Well, It depends on the application development. If product develop the objects with certain unique properties than identifying object will be easy by that property. But in practical world, that seldom happens. So, you need to carefully combine sets of properties to identify the objects. It is on the UFT programmer to set specific properties in particular category e.g. assistive, mandatory etc.

Aadhya: wow! So you mean I am the boss and can set any property to mandatory or assistive.

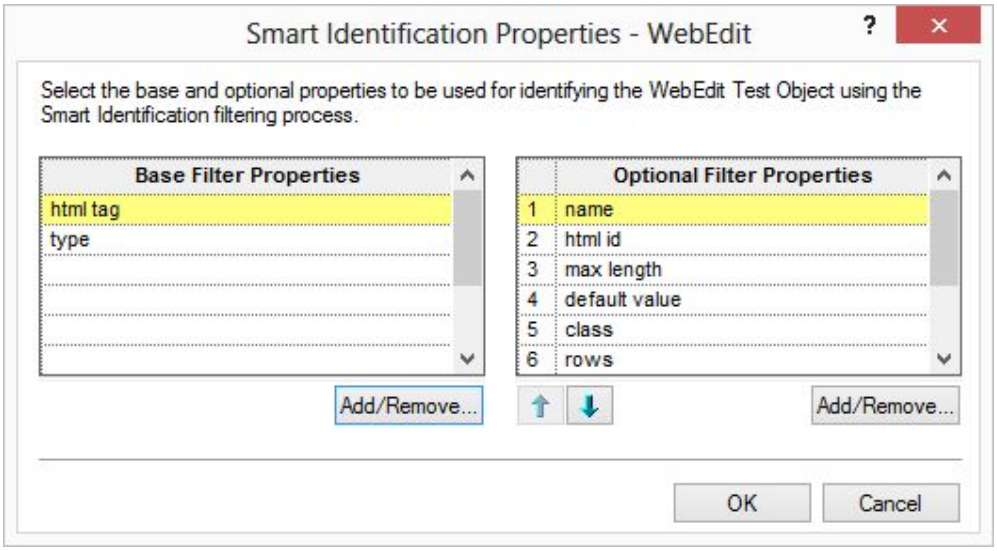
Aric: Yes| you are☺. But you should do it according to your application. Any misconfigured setting will lead UFT to record set of property that will eventually fail to recognize object during run-time. By default UFT provides default set of object identification properties and in most case they are apt to distinguish objects. These properties can be configured* from UFT Object Identification wizard.

STEP1: Configure Object identification properties from Object Identification wizard

Navigate to "FirstTest" in UFT | Tools | Object Identification | Environment 'Web' | WebEdit



Smart Identification Base filter and optional filter properties.



*ConfuSense: We need to select the property based on the way our application developed. If development team provide certain set of property for object then we should configure UFT accordingly.

2.3 Adding Object to Object Repository

Object can be added to the OR using one of the two methods:

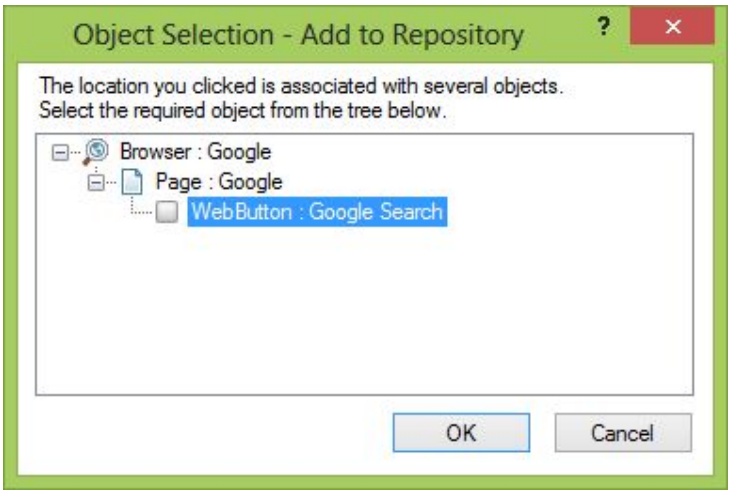
1. By recording interaction against the application under test
2. By manually adding one or more objects.

Object can be manually added to the OR by clicking on Add Objects button and then clicking on the object that needs to be added.

Note: In case the object we want to add appears after a mouse click, then press and hold the CTRL key prior to that mouse click. This temporarily disables the objects selection mode and allows us to perform mouse click operation to navigate. Then, when we are ready to add the object to OR , release the control key and click on the object.

In case we need to switch between applications, first press CTRL + ALT to disable the object selection mode. Then we can switch between different application and use key combinations like ALT + TAB etc. Once done, press the CTRL + ALT key to enable the object selection mode and add the object.

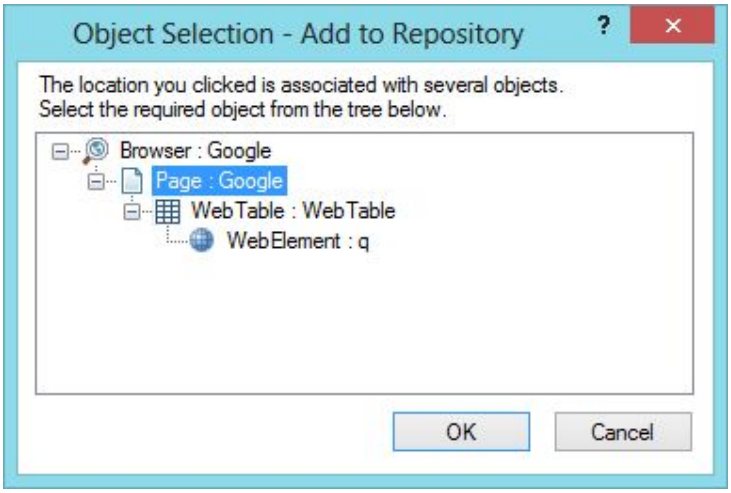
Once the object is selected UFT displays the object selection windows

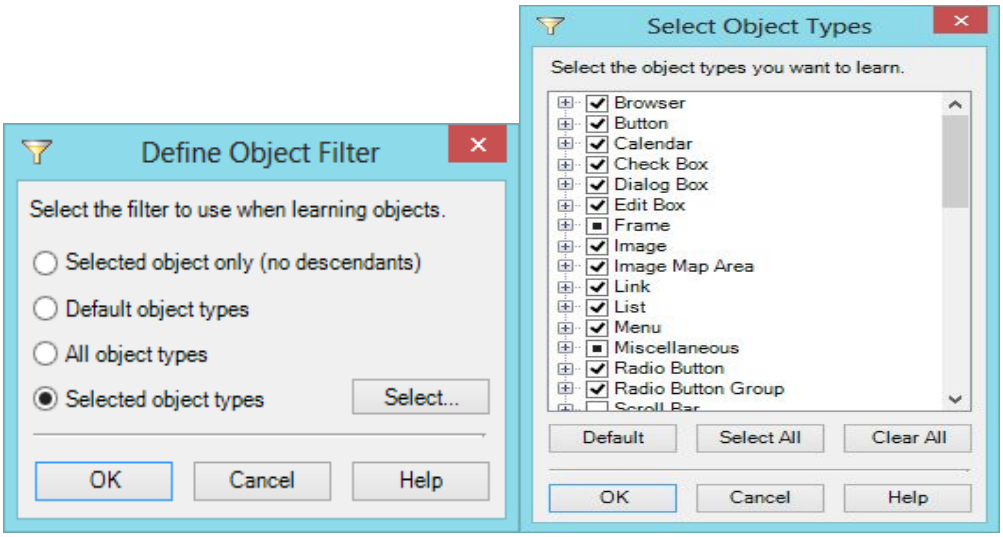


The Object selection window displays the complete hierarchy of the objects on the webpage. Select the objects which need to be added and click the OK button.

Note: The object hierarchy displays might not be the same the one recorded in the OR. UFT only keeps the hierarchy which is necessary for it to identify the object.

If we select the page object and continue, UFT will ask if we want to add all its child objects.





Selecting the select object type's radio button will give the choice to add specific type of object on the page.

Note: UFT does not add hidden objects present on the page.

4.0 VBScript and UFT

Let's explore the possibility to enhance out recorded test script.

Aadhya: I understand the UFT record the steps with object hierarchy and perform the available operations on that. For e.g. If I want to put "Krishna Wikipedia" in Google edit box than UFT will capture object, keep it in OR and record step in the editor.

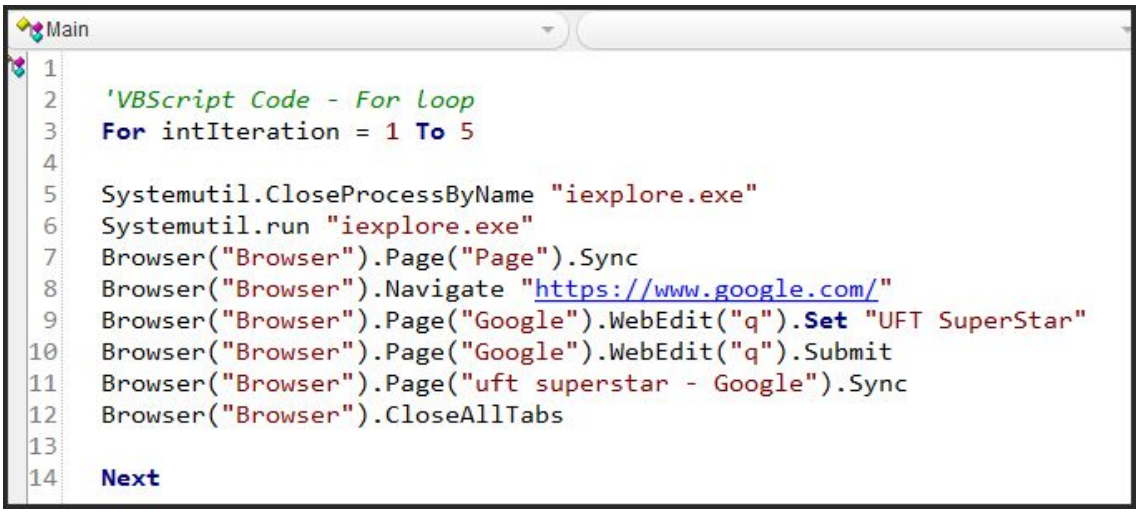
```
Browser("Browser").Page("Google").WebEdit("q").Set "Krishna Wikipedia"
```

But this is not VBScript. Aric, you said that UFT use VBScript to record the test steps.

Aric: Well catch Aadhya! UFT IDE understand VBScript and standard application objects for the specific add-in. So In this case, web Add-in will enable UFT to record/understand all web objects and supported operations. And each object support different operation.

STEP: Inserting VBScript logic in UFT to run same code five times

Open "FirstTest" in UFT | Modify test as given below



Now suppose you want to run the same test for 5 times so you can modify this script with VBScript logic. VBScript provide for…loop to iterate logic. In same way you can modify test with immense possibilities to do tasks like managing datasources (database, excel, files etc.), system settings, process, registry, browser settings and many more things. We will discuss VBScript in detail

Aadhya: Ok! I understood that UFT can use UFT recording code and enhance script by VBScript.

5.0 Understanding UFT IDE

Software occupation and buzz words go hand in hand. You may encounter many words which looks little vague but established as known terms for example GUI,OR, DP, DOM. IDE stands for integrated development environment. Net Beans and Eclipse are very known Java IDEs. Microsoft provides visual studio. Most of IDEs come with free and paid versions. IDE provide the facilities to develop computer programs. An IDE normally provide editor to write code, debugger, compiler or interpreter and output console. IDE sometime also assist by showing all available option for particular steps i.e. Intelli-sense coding features by pressing "." After the statement.
Now UFT also pass this test and comprises many facilities to debug, coding, output and report generation.

Exercise 1.2 (of 10-STEPS) – Explore UFT IDE

This exercise will explore the UFT menu options, configurable settings wizard, help and datatables.

STEP1: UFT Menu Options New Test

Open UFT | New | GUI Test with Name "ExploreUFT" | Create | Press {F6} or Record | Click on Internet explorer | Google.co.uk | Type "Krishna Wikipedia" in Edit Box | Enter | click on wikipedia link | Close Browser | Stop recording by clicking stop | Save

STEP2: Export test in .Zip File

Navigate to "ExploreUFT" in UFT | File | Export Test… | Zip file path "C:/Test" | OK | Navigate to "C:/Test" | unzip "ExploreUFT.zip" | Close

STEP3: Insert Code Snippet in script

Navigate to "ExploreUFT" | Edit | Code Snippet

STEP4: Reset Window Layout

Navigate to "ExploreUFT" | View | Reset Window Layout

STEP5: Configure Test Settings for specific test

Navigate to "ExploreUFT" | File | Settings…

STEP6: Options Settings to configure UFT IDE settings for all tests e.g. run sessions and other general settings

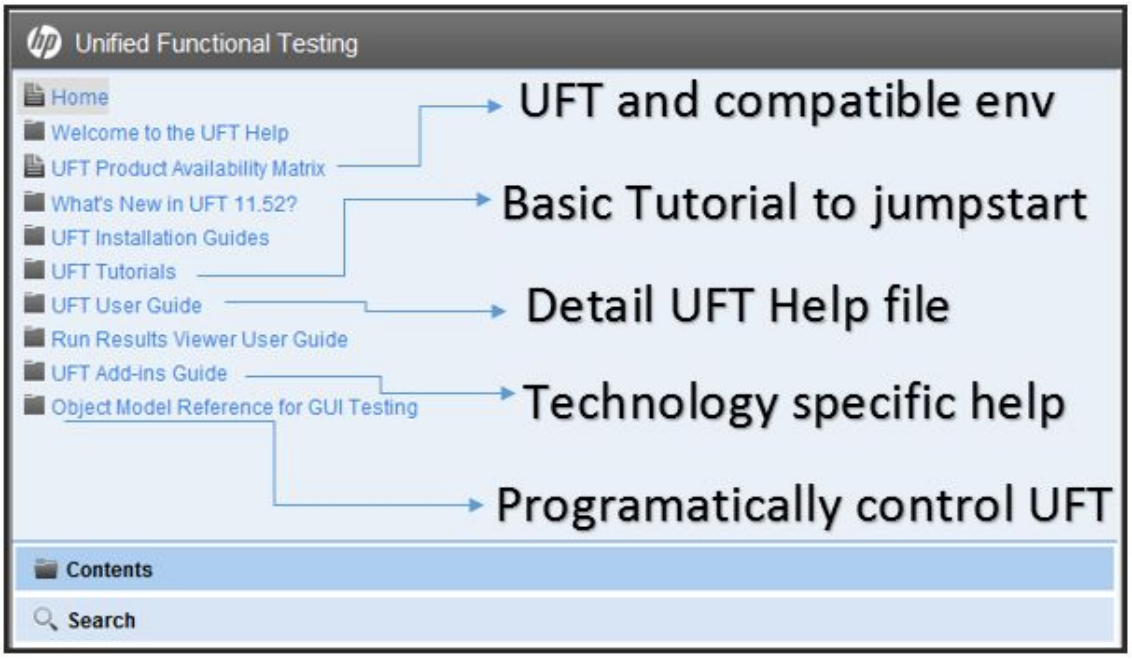
Navigate to "ExploreUFT" | Tools | Options… | Run Session | Deselect "View results when session ends" | OK

STEP7: Knowing about your HP UFT

Navigate to "ExploreUFT" | Help | About HP Unified Functional Testing | Detailed Info | System Info | Close | Licence… | Close | OK

STEP8: Help Guides, Index and Search any topic with keywords

*Navigate to "ExploreUFT" in UFT | Help | HP Unified Functional Testing He*lp

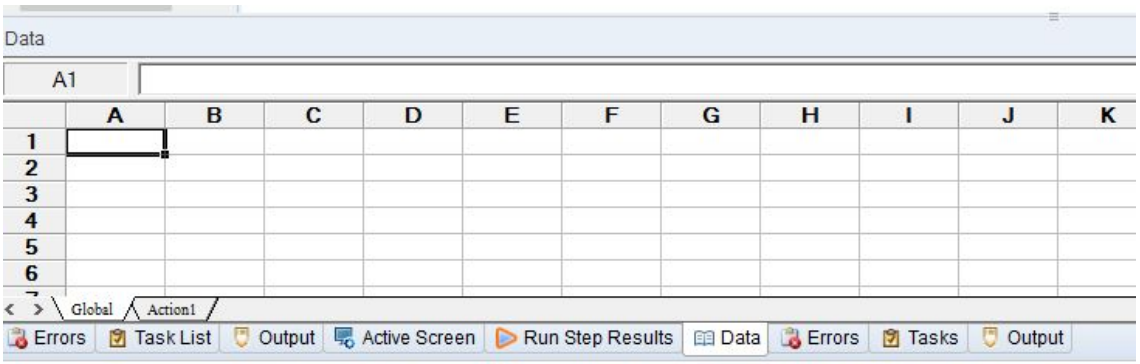


STEP9: Advance GUI Test Specific Help File

Navigate to "ExploreUFT" in UFT | Help | UFT GUI Testing Advanced References Help

STEP10: DataTable, Console and Active Screen Window

Navigate to "ExploreUFT" in UFT | View | Data | Output | Active Screen



Concept of Exercise 1.2 (of 10 STEPS) – Explore UFT IDE

In STEP1, UFT Menu contains all the UFT related features. In STEP2, we can export test in zip file. In STEP3, UFT script can be modified with decision and loop statements. In Step4, UFT can be reset to default view. In STEP5, we can configure test specific settings while in STEP6, we can set overall UFT specific settings. In STEP7, UFT provides information about version, licence and other crucial information related to UFT. In STEP8 and STEP9 tells about help section. In STEP10, UFT provide options for datatable and active screen.

DataTable used for provide the data to script. It looks like MS-excel sheet and have many similarity with excel but it is UFT object.

When we record the UFT test, UFT stores screenshot of application during that time. We need to put cursor in script at the step for which we want to verify the screen. These screenshots (known as Active screen) provides many other functionality other than display of the screenshots.

Key Learning of Exercise: UFT IDE provides comprehensive development environment. It provides user guide, datatable, test setting wizard etc.

6.0 Enhance Test Script to interact at run-time

UFT programmer need to communicate with script during run-time. Either script need to provide data or show current variable values in script.

Exercise1.3 (of 4-STEPS) – Interact with Script at run-time

This exercise will create test and ask user to provide input data and pop-up the total time script takes for whole execution.

STEP1: Create a test in UFT

Open UFT | New | GUI Test with Name "InteractUFT" | Create | Press {F6} or Record | Click on Internet explorer | Google.co.uk | Type "Krishna Wikipedia" in Edit Box | Enter | click on wikipedia link | Close Browser | stop | Save

STEP2: Create a test in UFT

Navigate to"InteractUFT" in UFT | Modify test as given below | Save | Press F9 at 2nd line| Run

STEP3: Verify UFT Console

Navigate to"InteractUFT" in UFT | View | Output | Select "Print Log"

STEP4: Search UFT helpful to find the function specification

Navigate to"InteractUFT" in UFT | Help | UFT GUI Testing Advanced References Help | Index | "InputBox" | Click on "InputBox Function"

```
InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context])
```

Concept of Exercise1.3 – Interact with Script at run-time

In STEP1, test create to search Wikipedia article on Krishna. In STEP2, timer returns the "total seconds" passed since midnight. We created two variable at line-1 and Line-14 to store total seconds passed after midnight. So, the value of (endTime - startTime) will return the total seconds spend during Line-2 to Line-13.

We are closing internet explorer process in Line-2 and reopening internet explorer in Line-3. SystemUtil is a utility and reserved object i.e. they do not store in object repository. SystemUtil used to control processes. It can block/unblock keyboard and mouse inputs, open/close any application or process (by name, id or Hwnd) by specifying the correct path. Sometime when we record test on

any open browser, we may get first line with sync statement after recording. We should remove it and open application with “SystemUtil.Run” statement.

```
1   Browser("Browser").Page("Page").Sync
```

VBScript MsgBox invokes pop-up and can show value passed in it. Operator “&” used to concatenate two values in VBS. In this example it will add static string “Total Time” with variable value (endTime - startTime).

```
15          Value1    "&" concatenate two strings    Value2
16   MsgBox "Total Time = " & (endtTime - startTime)
```

In STEP3, UFT will print value in UFT console. Print is UFT specific.

In STEP4, InputBox can take user input at run-time. We can search various function specification in Index at help file while “search” will list down all the topic with the given keyword. Input box can take total seven parameter out of which all parameter in parenthesis ([]) are optional. We have provided the three parameter in our example.

```
InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context])
          1          2          3
InputBox ("Provide value",  "Google Search”, “Krishna wikipedia")
```

Key Learning of Exercise: We need to write logic in the script to drive business flow. User can provide input by using “inputbox” function or show value in message box by sending variable in MsgBox function. These both are VBScript functions. UFT also provides print statement to send information on output window in UFT. Operator “&” concatenate two variables (or strings).

Concept of Exercise 1.3 – END

Aadhya: I understand the UFT IDE and the way I can interact with it. But while I am creating script and running them they are sometime raising errors.

Aric: There may be many reasons for UFT to fail. Common reasons are synchronization, object non identification, duplicate object, system settings not compatible with UFT, Browser and OS version not supported by UFT, any environment issue e.g. Pop-up, alert at run time etc.

Aadhya: ok, I am feeling that my UFT is fast then browser

Aric: This is synchronization issue. This topic will be covered in detail but time being you can use wait statement wherever you feel synchronization issue example wait (5)

7.0 Solution in UFT

Exercise1.4 (of 4-STEPS) – Solution in UFT

This exercise will create test and ask user to provide input data and pop-up the total time script takes for whole execution.

STEP1: Create a test in UFT

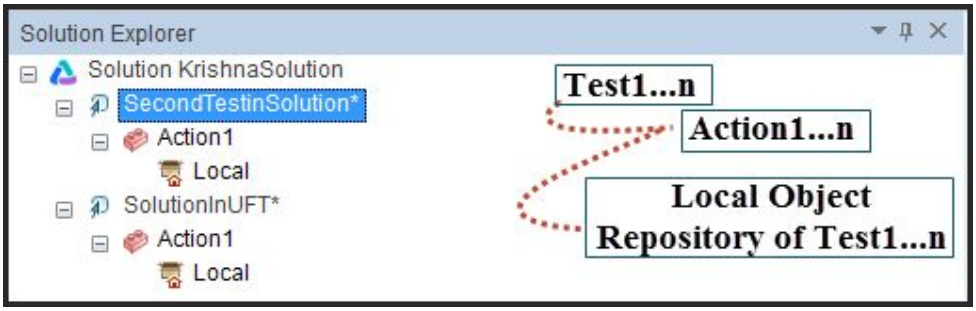
Open UFT | New | GUI Test with Name "SolutionInUFT" | Create | Press {F6} or Record | Click on Internet explorer | Google.co.uk | Type "Krishna Wikipedia" in Edit Box | Enter | click on wikipedia link | Click on "International Society for Krishna Consciousness" Link | Close Browser | stop | Save

STEP2: Create another test in same solution UFT

Navigate to UFT | Right click on Solution | Add | Add New Test… | GUI Test with Name "SecondTestinSolution" | Add | Double click on "SecondTestinSolution" | Press {F6} or Record | Click on Internet explorer | Google.co.uk | Type "Krishna Wikipedia" in Edit Box | Enter | click on wikipedia link | Click on "Bhagavad Gita" Link | Close Browser | stop | Save

STEP3: Save the solution

Navigate to UFT | Click on "Solution Untitled" in UFT | File | "Save Untitled As" | "KrishnaSolution" in "C:\Test" | Right Click on "Solution KrishnaSolution" | Expand All | Save



STEP4: Run specific test from solution

Navigate to "Solution KrishnaSolution" UFT | Press F5 to run test | Select the test name "SolutionInTest" from drop down | Expends Options | Provide "C:\Test\Res2" in "New run results folder" | Run | View | Last Run Results | ALT + X + V | Close

Concept of Exercise1.4 – Solution in UFT

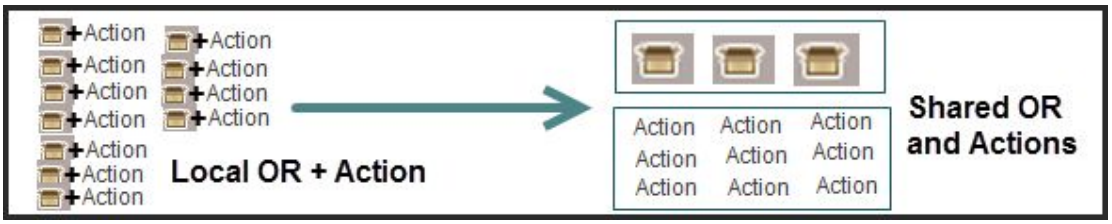
In STEP1, we have created a test to search "Krishna Wikipedia" on Google and navigate to wiki link. In STEP2, we have again created second test. This is little non-traditional then previous UFT versions where user can only able to create single test during test design. In STEP3, we saved solution with two tests. In STEP4, we are running the solution.

Key Learning of Exercise: UFT IDE is very different from its previous versions. It is more streamline to the existing popular IDE. UFT IDE tries to organize test script, resources, parameters, properties more effectively. Solution is the umbrella of all these IDE features.

Exercise1.4 – Ends

8.0 Shared Vs. Local Object Repository in UFT

When we record script UFT create a object repository. This is local object repository. So if we record 100 script we will end-up with 100 ORs. Even all these OR may have nearly similar objects. When UFT replay these OR need to load in memory (RAM). If there is a quick test runs then loading and unloading will be a performance overhead as well. Few Shared object repositories can solve these issues. Shared object repositories can associate with tests and remove the dependency of loacal OR with script.



Exercise 1.5 (of 5-STEPS) – Understanding Shared and Local repository

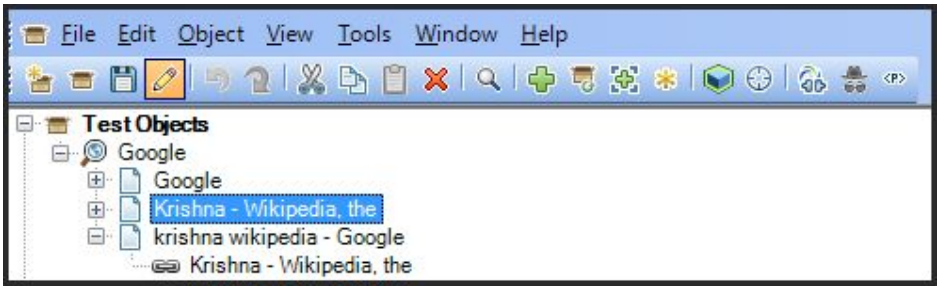
STEP1: Create a Folder for the Object Repository (SOR)

Navigate "C:\Test" | Create a folder OR

STEP2: Create a Shared Object Repository (SOR)

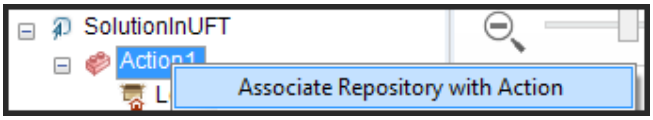
Navigate UFT | New | GUI Test with Name "SOR" | Create | Resource | Object Repository Manager… | Open Internet Explorer | Go to Google.co.uk | Navigate to UFT Object Repository Manager Wizard | Click on | Click on Google page| Select "Page:Google" in Object Selection | OK | Select Object Type | Select | Select All | OK | Navigate to google | search "Krishna Wikipedia" | Enter | Navigate to UFT Object Repository Manager Wizard | Click on | Click on "Krishna - Wikipedia, the free encyclopedia" link | Select link in OR Wizard | Navigate to google | Click on

"Krishna - Wikipedia, the free encyclopedia" link | Navigate to UFT Object Repository Manager Wizard | Click on  *| Click on Wikipedia page | Select | Select "Page: Krishna- Wikipedia, the free encyclopedia" in Object Selection | Select Object Type | Select | Clear All | Select only link | OK | Go to File in Object Repository Manager Wizard | Save As… | GlobalSOR1.tsr | OK*



STEP3: Remove Local repository and link test to Shared object repository in first Test

Open "KrishnaSolution" in UFT | Right click on Solution | Expand all | Right click on Action in "SolutionInUFT" | Associate Repository with Action | Navigate to "C:\Test\OR\ GlobalSOR1.tsr" | Open | No | Right click on Local | Open Repository | Select Local Object | Delete all local Objects | OK | Close OR | Save



STEP4: Remove Local repository and link test to Shared object repository in second test

Navigate to "KrishnaSolution" in UFT | Right click on Solution | Expand all | Right click on Action in "SecondTestinSolution"| Associate Repository with Action | Navigate to "C:\Test\OR\ GlobalSOR1.tsr" | Open | No | Right click on Local | Open Repository | Select Local Object | Delete all local Objects | OK | Close OR

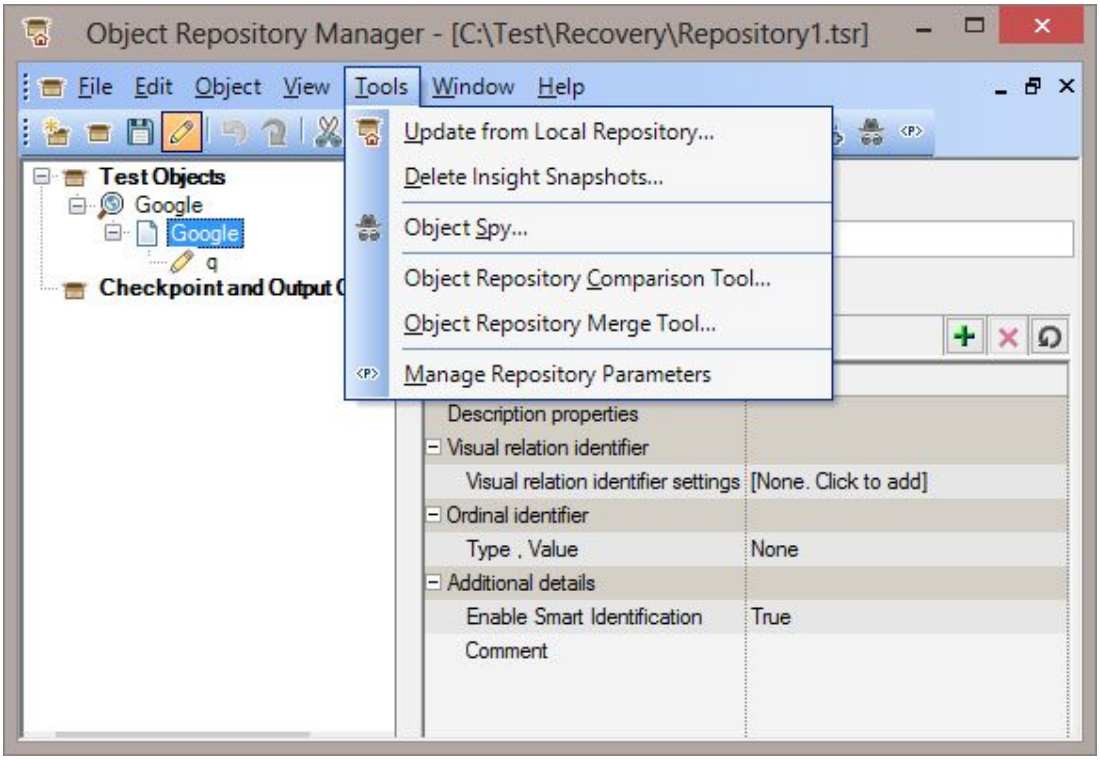
STEP5: Run Test

Navigate to "KrishnaSolution" in UFT |Press F5 to run | Run

Concept of Exercise 1.5 – Understanding Shared and Local repository

In STEP1, We are creating a folder to save our shared object repository. In STEP2, We are adding object in repository by using UFT Object Repository Manager Wizard. We have not recorded anything but picking the object to make the object basket. This is like plucking the flower and putting in separate basket. Object Repository Manager Utility provides a way to manage object repositories. In STEP3, we are removing local OR and associating shared OR. In STEP4, we are removing local OR and associating shared OR again. We should remember that it is not necessary to remove local repository as UFT can have shared (read-only) and local repository associated with single test.

Object Repository Manager (ORM) can export, import OR in XML format. ORM provides object repository comparison tool and merge tool. Merge tool merge two ORs and resolve the conflict of the similar objects (hierarchy, name and properties). ORM is very important and helpful tool to manage the Shared ORs.



When should use Shared and Local OR?

This is very crucial design consideration that when we should use which kind of OR. These are the few points that need to understand before considering any option. Remember these points are not mutually exclusive but provide collective insight to take decision.

Category	Local OR	Shared OR
Modularity	Test with single or few action	Multi Action or test suite based on framework
Number of Object or Test	Very few test or objects in testing journey	Automation suite with many tests and large number of test objects.
Modification need	Frequently not required	Frequently need to modify objects by repository manager

Learning Stage	New to UFT automation and application.	Good understanding of UFT and application
Co-working	Few people working in team	Multiple people working in team

Exercise - Associate Shared Object Repository

STEP1: Create a test in UFT

 *Open UFT | New | GUI Test with Name "MySharedOR" | Create | Press {F6} or Record | Click on Internet explorer | QuickTestPro.co.uk | Click Home link | Close Browser | stop | Save*

STEP2: Create two shared ORs with one Local OR

 *Open IE | Navigate to "QuickTestPro.CO.UK" | Navigate to UFT | Resource | Object Repository Manager | Add Object | | Select Page Object | Default Object Type | OK | Save | Save file in OR folder "C:\Test\OR\QTPUK" (Create OR folder if not exist) | OK | navigate to ORM | File | New | Add Object | Click on "Krishna Wikipedia" page | Select Page Object | Default Object Type | OK | Save | Save file in OR folder "C:\Test\OR\QTPUK2" | Close*

STEP3: Compare both Shared ORs

 *Navigate to UFT | Resource | Object Repository Manager | Tools | Repository comparison | Select "C:\Test\OR\QTPUK" and "C:\Test\OR\QTPUK2" | OK*

STEP4: Merge both Shared ORs

 *Navigate to UFT | Resource | Object Repository Manager | Tools | Object Repository Merge Tool | Select "C:\Test\OR\QTPUK" and "C:\Test\OR\QTPUK2" | OK | Verify Merge Summary | Save | MergeQTPUK | Save*

STEP5: Associate Shared OR with Test and verify Local and Shared OR

 *Navigate to UFT | Resource | Associate Object Repository | + | Select "C:\Test\OR\ MergeQTPUK" | Action 1 | > | OK | Go to Menu Run | Step Into (or F11) | Resource | Object Repository | Verify Local and Shared OR | Run*

Concept of Exercise - Associate Shared Object Repository

IN STEP1, we recorded script on QuickTestPro.co.uk. In STEP2, we create two shared object repositories. In STEP3, we are using shared object repository manager's comparison functionality to compare two OR (.tsr) files. In STEP4, we are again exploring merge functionality. In STEP5, we are associating shared or with test. Manipulating shared OR by object repository manager does not attach OR to test. We must need to associate the shared OR with actions explicitly.

Comparison and Merge functionality is very helpful features of share object repository manager. If you have multiple OR then analysis and conflict can be resolved by them.

Key Learning of Exercise: Shared object repository provides way to modularize ORs. We can associate more than one OR with test. Object resource manager is provides advance option to manage, update, delete, compare and marge object repositories.

Recording Mode in UFT

UFT records the user's action. This recoding can be done in many modes. Suppose you want to automate a tic-tac-toe application where you need to use mouse to sketch cross or right sign. Also after winning the game you need to provide autograph. These kind of special users' action required different kind of recording mode then just "fill-click-proceed" normal mode.

Exercise: Understanding UFT recording mode

STEP1: Record test with different recording mode

 *Navigate to UFT | Test | New | "Recording Mode" in UFT | Create | Record or F6 key | Open Internet explorer | Gmail.com | Provide User name and password | Click Sign in | Change mode to Analog mode | Click on*  *| Click on "Cozy" | Click on "Compact" | Click on "Comfortable" | Click Inbox | Change Low-Level mode | Click on* 

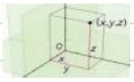
| Click on Inbox | Change mode to Insight recording mode | Click on Gmail icon | Click on mail Search button | Change to default mode | logout | Close browser | stop recording | Save | Tools | Options | GUI Testing | Test Runs | Provide 2000 millisecond in Delay each step execution in Normal Mode | OK | Run by F5 key | Tools | Options | GUI Testing | Test Runs | Provide 0 millisecond in Delay each step execution in Normal Mode | OK | Save

STEP2: Explore the result test with different recording mode

 *Navigate to UFT | View | Last Run Result | ALT + V + X | Verify the result of recording mode*

Concept of Exercise: Understanding UFT recording mode

In STEP1, we are recording test with analog and Insight recording modes. Analog mode created a track which contains mouse and keyboard operations. You cannot see the steps in analog mode. Low level recoding is like a dart game where you perform action on particular x and y coordinates. Insight mode is new mode in UFT. This recoding mode is based on the appearance of the application. It records application's object based on their looks and neighbours.

RECORDING			
	1	Insight Recording .InsightObject(Google Search).click	Recognizes Object based on their appearance 
	2	Analog Recording .RunAnalog "Track1"	Exact mouse and keyboard operations 
	3	Low-Level Recording .Click 536,261	Exact coordinates of any object 
	4	Normal Recording .Image("SignIn").Click	Default mode that by default you record in UFT

Analog Mode

This mode is useful for recording operations that cannot be recorded at the level of an object, for example, sketching or a signature by dragging mouse. You can mix the normal mode and analog mode so it provide flexibility to test end to end scenario with the part necessary to perform tricky mouse dragging operations.

Analog mode helps to record the scenario where user need to record mouse movement. For example, a mouse signature or working with drawing applications that create images by dragging the mouse. The track records operation relative to desktop screen or application window

```
Desktop.RunAnalog "Track1" ‘Desktop
Window("KrishApplication").RunAnalog "Track2" ‘ Application window
```

There will be no OR update for analog mode. The track file called by the RunAnalog method contains all your analog data and is stored with the current action.

Low-Level Mode

Low level mode enables recording when UFT face difficulty to recognize the specific object (e.g. customize objects). This mode records at the object level and records all run-time objects as Window or WinObject test objects.

When should choose Low level mode?

low-level recoding	Description
Unsupported Environment	Low level recording mode helps recording on objects not supported by UFT. If the appearance of the objects might change, but their location will not then use low level recording mode. If appearance will not change but location changes then we should use insight recording mode.
Need to Record Exact Location	Use low-level recording for when you need to record the exact location of the operation on your application screen. If the location of the object is important to your test or scripted component, switch to low-level recording to enable UFT to record the object in terms of its x- and y- coordinates on the screen.

Whole family is window family	UFT records all parent level objects as Window test objects and all other objects as WinObject test objects. They are displayed in the Active Screen as standard Windows objects.
Method supported in low-level	WinObject test objects: Click, DblClick, Drag, Drop, Type Window test objects: Click, DblClick, Drag, Drop, Type, Activate, Minimize,Restore,Maximize
Shows step in script	Unlike analog mode, each step recorded in low-level recording mode is added to the test and can be viewed in the Keyword View and Editor.
Different Characters	Low-level recording mode is not fully supported for multibyte character input (e.g. Korean, Japanese etc.)
Not completely Reliable	Steps recorded using low-level recording mode may not run correctly on all objects.
More Disk Space	Low-level recording requires more disk space than normal recording mode.

Insight mode

This is new recording mode in UFT. Insight mode recognizes objects based on their appearance, and not their native properties. So, Automation can be possible for the applications based on the non-standard test objects or even from a remote computer running a non-Windows operating system.

Mixing the recording mode is possible and you can switch to Insight recording in the middle of a recording session for specific steps. When created, all Insight test objects are named **InsightObject**, with an incremental suffix added to test object. Insight recording requires more disk space (configurable) than normal recording mode, because of the test object image and the snapshots stored with the test object.

Summary

Aadhya: Aric! I understood that UFT store the object in OR. During run time, UFT recognize the object with mandatory, assistive and ordinal properties configured in Object Identification settings. Every object has different set of operations in UFT. We can modify the UFT script by using VBScript logic in that.

Aric: Yes! We will continue to learn more advance topic to enhance our test.