Chapter -8

UFT – Managing Test 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

In this chapter we will discuss the way to enhance and manage our tests. We will learn Utility Object, File system and Dictionary Object in this chapter

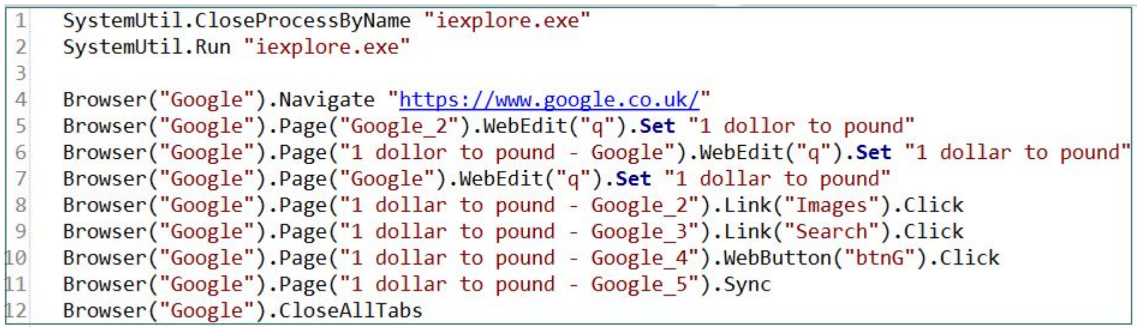
1.0 Working with Files

Automation interact with files to in/out the data and drive the test. VBScript provides FileSystemObject to create, open, read, write, amend, delete and copy files . FileSystemObject not only manages files but also handles folder, drive and text streams.

Exercise: Reading and Writing Files by VBScript

Step1: Create a Google Search

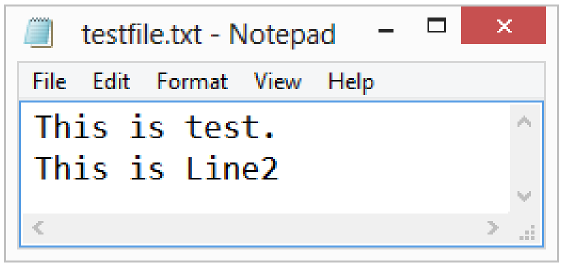
Open UFT | New |Test | Select GUITest with "UFTandFSO" | Create | Record | SelectFirst radio(run on any browser) | Ok | Internet Explorer | Google.co.in | Search with "1 dollar to pound" | Click Search button | Click on | Close IE | Stop |Modify Script as given below | Save



```
SystemUtil.CloseProcessByName "iexplore.exe"
SystemUtil.Run "iexplore.exe"

Browser("Google").Navigate "https://www.google.co.uk/"
Browser("Google").Page("Google_2").WebEdit("q").Set "1 dollor to pound"
Browser("Google").Page("1 dollor to pound - Google").WebEdit("q").Set "1 dollar to pound"
Browser("Google").Page("Google").WebEdit("q").Set "1 dollar to pound"
Browser("Google").Page("1 dollar to pound - Google_2").Link("Images").Click
Browser("Google").Page("1 dollar to pound - Google_3").Link("Search").Click
Browser("Google").Page("1 dollar to pound - Google_4").WebButton("btnG").Click
Browser("Google").Page("1 dollar to pound - Google_5").Sync
Browser("Google").CloseAllTabs
```

STEP2: Create and Write text File by QTP

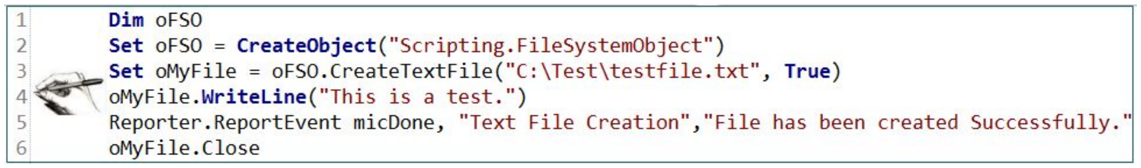
Navigate to "C:\Test" | Create a text file "testfile.txt" | Write two lines "This is test." and "This is Line2" | Save | Close



```
This is test.
This is Line2
```

STEP3: Create FileSystemObject in Script

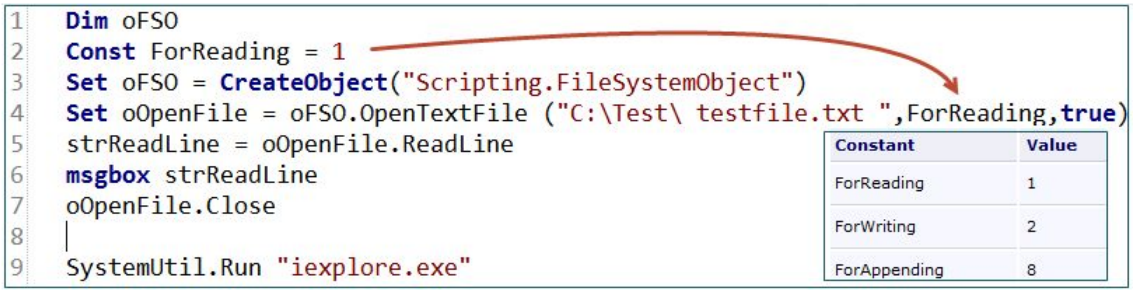
Navigate to "UFTandFSO" in UFT | Modify script by adding following Lines in beginning of the script |Design | Check Syntax or "CTRL + F7" | Run | Save | View | Last Run Result |Press three keys simultaneously "ALT +V + X" | Verify the user define reporting message to UFT report.



```
1    Dim oFSO
2    Set oFSO = CreateObject("Scripting.FileSystemObject")
3    Set oMyFile = oFSO.CreateTextFile("C:\Test\testfile.txt", True)
4    oMyFile.WriteLine("This is a test.")
5    Reporter.ReportEvent micDone, "Text File Creation","File has been created Successfully."
6    oMyFile.Close
```

STEP4: Read File by UFT

Navigate to "UFTandFSO" in UFT | Modify script by removing 1 to 6 lines and adding following Lines |Design | Check Syntax or "CTRL + F7" | Run | Save



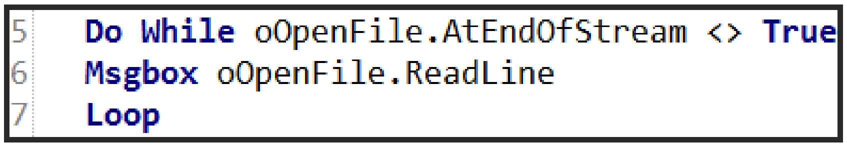
```
1    Dim oFSO
2    Const ForReading = 1
3    Set oFSO = CreateObject("Scripting.FileSystemObject")
4    Set oOpenFile = oFSO.OpenTextFile ("C:\Test\ testfile.txt ",ForReading,true)
5    strReadLine = oOpenFile.ReadLine
6    msgbox strReadLine
7    oOpenFile.Close
8    
9    SystemUtil.Run "iexplore.exe"
```

Constant	Value
ForReading	1
ForWriting	2
ForAppending	8

STEP5: Read Complete text file with AtEndOfStream

We can iterate the complete text file. Text file ends when script finds AtEndOfStream= True. So, the following loop will read each line in text file.

Navigate to "UFTandFSO" in UFT | Modify line 5 by replacing following lines |Design | Check Syntax or "CTRL + F7" | Run | Save



```
5    Do While oOpenFile.AtEndOfStream <> True
6    Msgbox oOpenFile.ReadLine
7    Loop
```

ConfuSense – CONST is a constant variable that cannot be change during the test execution. If you try to change the value VBS engine will throw error.

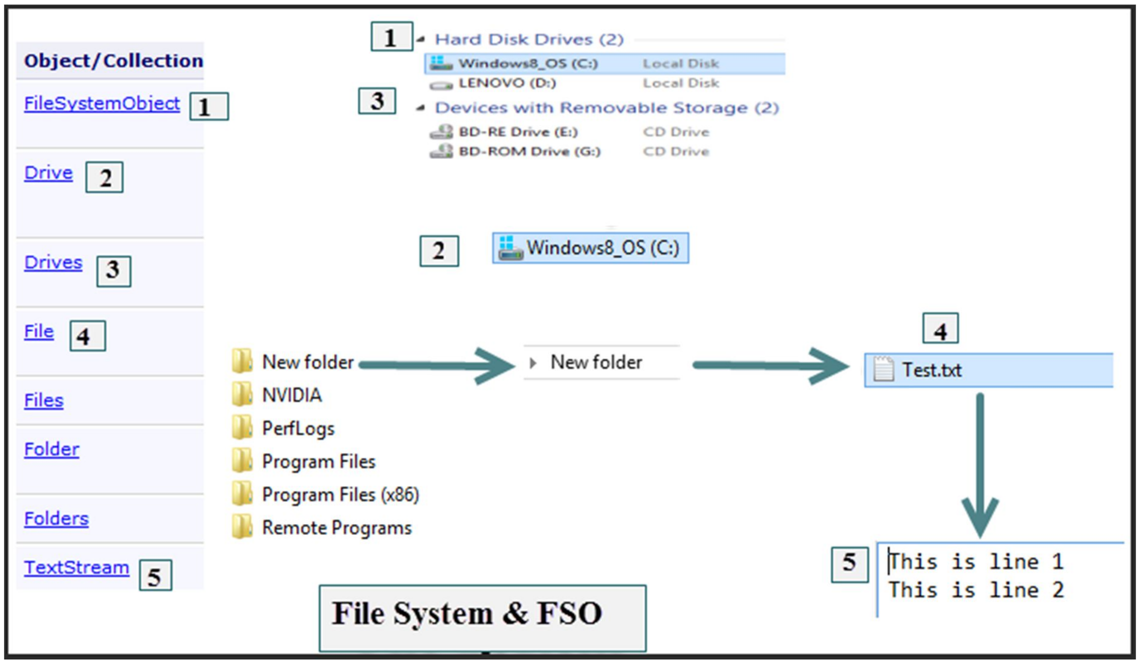
Concept of Exercise: Reading and Writing Files by VBScript

In STEP1, we have created a Google search test for converting "1 dollar to pound". In STEP2, we create a text file with 2 lines. In STEP3, we coded filesystemobject (FSO) to create and write a line in file. In STEP4, we read text file. The "Opentextfile" method support to open text file in read,write and append mode. We need to provide numeric value (1- Read, 2-Write and 8-Append). Third optional argument "true" creates new file while "false" will not allow new file creation. In STEP5, we are reading complete file line by line. The "AtEndOfStream" will keep eye on the textstream and will return false until we exhaust with last line in text file.

FileSystem Object:

Filesystem object provide provides access to a computer's file system.

Compute file system contains the Items as shown in figure Do not confuse with drives. It is from C: to G:



Filesystemobject provide very powerful facility to manage file system. We first need to create an object of Filesystemobject by CreateObject and then we can use various method, properties* and other sub-objects to manage the Filesystem.

This example shows the steps to read a text file

1. Create FSO Object by **Set** oFSO = **CreateObject**("Scripting.FileSystemObject")
2. OpenTextFile– ForReading, ForWriting or ForAppending
3. ReadLine, WriteLine to read or write in file.

ConfuSense – Object use dot (.) operator for property as well as method. Property set or return the value while method call a function to perform something e.g. Set folder = fso.GetFolder("c:\") is calling GetFolder method while fldr.Name is calling the property "Name" of that folder. You should see the specification in helpfile to find out the full object supported methods and properties.

Filesystem comprises the drives, folders and files with specific permission to the user or group. There are various type of drive and files depending on their specification. FileSystemObject along with other objects e.g. Drive object, file object, folder object and textstream object provide various method and properties to manage Filesystem.

Exercise: Exploring FileSystem Object

To program with the **FileSystemObject** (FSO) object model:

Step 1: Use the **CreateObject** method to create a **FileSystemObject** object.

```
Set oFSO = CreateObject("Scripting.FileSystemObject")
```

Step 2: Create Object for Driver, Folders or Files.

```
Set dc = fso.Drives
```

Step 2: Access the object's properties.

```
For Each d in dc
MsgBox d.DriveLetter
Next
```

ConfuSense – The FSO object model is contained in the Scripting type library, which is located in the Scrrun.dll file. Therefore, you must have Scrrun.dll in the appropriate system directory to use the FSO object model.

Concept of Exercise: Exploring Filesystem Object

In STEP1, we create an fso object (when we create an object is initialize with null). We can create more than one FSO object by using similar statement, if required. In STEP2, we created a drive object. In STEP3, we use for-each loop to iterate all drives. The "DriveLetter" is a property that return letter of the drives. In STEP2 we created the drive object similarly the following table depicts the supported properties and methods of all file system objects.

OBJECT	Description	Properties and Methods
Drive	Object. Drive , CD-ROM drive, a RAM disk etc.	AvailableSpace Property | DriveLetter Property | DriveType Property | FileSystem Property | FreeSpace Property | IsReady Property | Path Property | RootFolder Property | SerialNumber Property | ShareName Property | TotalSize Property | VolumeName Property
Drives	Collection of Drive	Count Property | Item Property

File	Object.	**Method**: Copy Method | Delete Method | Move Method | OpenAsTextStream Method **Properties** : Attributes Property | DateCreated Property | DateLastAccessed Property | DateLastModified Property | Drive Property | Name Property | ParentFolder Property | Path Property | ShortName Property | ShortPath Property | Size Property | Type Property
Files	Collection of files contained within a folder.	Count Property | Item Property
Folder	Object.	**Methods** Copy Method | Delete Method | Move Method | CreateTextFile Method **Properties** Attributes Property | DateCreated Property | DateLastAccessed Property | DateLastModified Property | Drive Property | Files Property | IsRootFolder Property | Name Property | ParentFolder Property | Path Property | ShortName Property | ShortPath Property | Size Property | SubFolders Property | Type Property
Folders	Collection of folder	Count Property | Item Property
TextStream*	Object. Allows you to read and write text files.	**Methods** Close Method | Read Method | ReadAll Method | ReadLine Method | Skip Method | SkipLine Method | Write Method | WriteBlankLines Method | WriteLine Method **Properties** AtEndOfLine Property | AtEndOfStream Property | Column Property | Line Property

ConfuSense – The FSO object model, which is contained in the Scripting type library (Scrrun.dll), supports text file creation and manipulation through the **TextStream** object.

A small framework with file system and UFT Utilities

Though VBScript program can enable us to do most of the automation task but UFT also provides in-built utility objects to accomplish most common automation task e.g. generate random number, file existence verification, opening and closing windows processes, skipping any intermittent occurring step in test script etc.

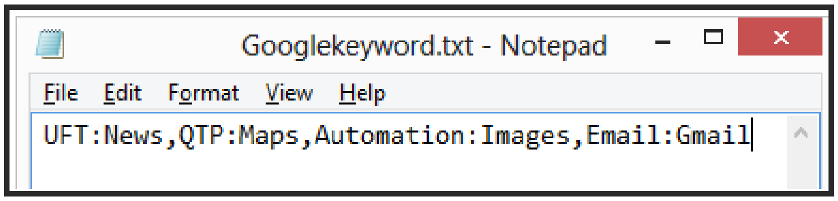
Problem statement:

We need to develop a test script where we can do the following

1. Create a text file with Google search keywords
2. Open internet Explorer and navigate to http://www.google.co.in
3. Verify text file existence before doing any operation
4. Search Google with keyword from File
5. Generate Random Number and add in search keyword
6. Modify test setting at run time
7. Monitor System Performance
8. Report Total time taken by script in UFT report
9. Capture Image before ending the test
10. Report should not show password given in script.

Step1: Create a text file with Google search keywords and link names

Navigate to "C:/Test/Data" | Create GoogleKeywords.txt text file| write "UFT:News,QTP:Maps,Automation:Images,Email: Gmail" in text file | Save Close



```
UFT:News,QTP:Maps,Automation:Images,Email:Gmail
```

Explain: In this file we created keyword:LinkOnGoogle pair. We will program in a way that script will pick this pair and search for keyword and click on the link name to find the result of that key.

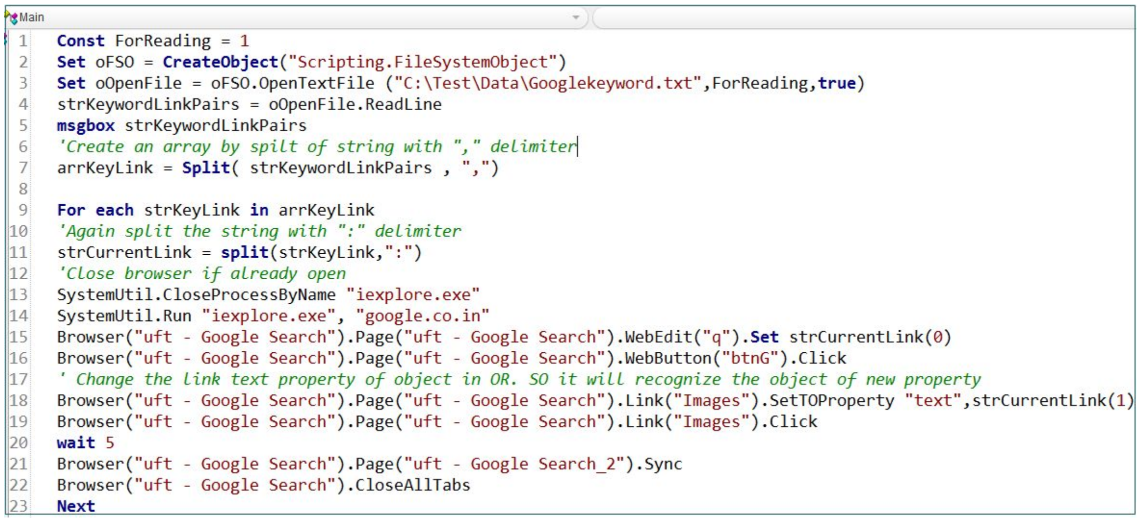
Step2: Record the Google Script

Open UFT | New |Test | Select GUITest with "UtilitiesUFT" | Create | Record | SelectFirst radio i.e. run on any browser | Ok | Internet Explorer | Google.co.in | "uft" in google Box | | Click Search | Go to recording bar and insert standard checkpoint | Click on Page | Select page from Hierarchy | OK | OK | Click on Images Link | Close IE | Stop | Save

Explain: we recorded test here. As of now script has all hard-coded values.

Step3: Parameterize the test script value from file with FileSystemObject.

Navigate to UtilitiesUFT in UFT | Modify script with the following code



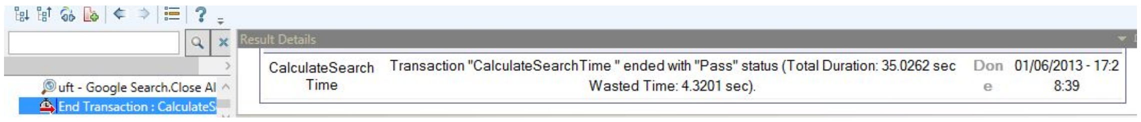
```
Const ForReading = 1
Set oFSO = CreateObject("Scripting.FileSystemObject")
Set oOpenFile = oFSO.OpenTextFile ("C:\Test\Data\Googlekeyword.txt",ForReading,true)
strKeywordLinkPairs = oOpenFile.ReadLine
msgbox strKeywordLinkPairs
'Create an array by spilt of string with "," delimiter
arrKeyLink = Split( strKeywordLinkPairs , ",")

For each strKeyLink in arrKeyLink
'Again split the string with ":" delimiter
strCurrentLink = split(strKeyLink,":")
'Close browser if already open
SystemUtil.CloseProcessByName "iexplore.exe"
SystemUtil.Run "iexplore.exe", "google.co.in"
Browser("uft - Google Search").Page("uft - Google Search").WebEdit("q").Set strCurrentLink(0)
Browser("uft - Google Search").Page("uft - Google Search").WebButton("btnG").Click
' Change the link text property of object in OR. SO it will recognize the object of new property
Browser("uft - Google Search").Page("uft - Google Search").Link("Images").SetTOProperty "text",strCurrentLink(1)
Browser("uft - Google Search").Page("uft - Google Search").Link("Images").Click
wait 5
Browser("uft - Google Search").Page("uft - Google Search_2").Sync
Browser("uft - Google Search").CloseAllTabs
Next
```

Explain: In this step we reads the keyword and link pair (in format of UFT:Images) and parameterize UFT script. Keyword will use for Google search box and Link will be used to click link on google home page. VBS split function splits the string with specified delimiter and return array with values. This is the step where our flow is driving from the text file i.e. data is driving our test.

Step 4: Calculate time in steps

4.1 Report time By UFT's Transaction utility

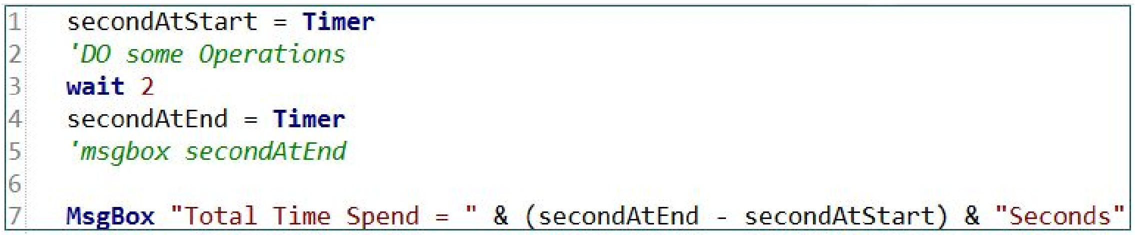
Navigate to UtilitiesUFT in UFT | Right Click on First line| Insert Step | Start Transaction | CalculateSearchTime | Verify step Services.StartTransaction " CalculateSearchTime " | Right Click on last line in Script | Insert Step | End Transaction | Select CalculateSearchTime | OK | Save | Run | View | Last Run Result | ALT + V + X | Navigate to "End Transaction : CalculateSearchTime"



CalculateSearch Time	Transaction "CalculateSearchTime " ended with "Pass" status (Total Duration: 35.0262 sec Wasted Time: 4.3201 sec).	Done	01/06/2013 - 17:28:39

Explain: Services.StartTransaction and Services.EndTransaction used to calculate and report the time between two steps during run time. StartTransaction must end with EndTransaction and they can appear between any steps in script. This is same concept used in performance testing tools e.g. Loadrunner.

4.2 Calculate time in Seconds by VBS Timer Function

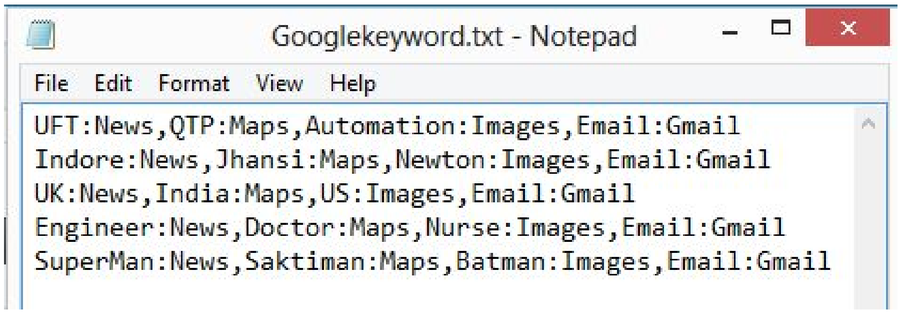


```
secondAtStart = Timer
'DO some Operations
wait 2
secondAtEnd = Timer
'msgbox secondAtEnd

MsgBox "Total Time Spend = " & (secondAtEnd - secondAtStart) & "Seconds"
```

Explain: We can also calculate the time using VBS timer function. Timer function return the second elapse after midnight. So if we catch the seconds in two variables at different steps and substract them subsequently then we can get the seconds difference between those steps.

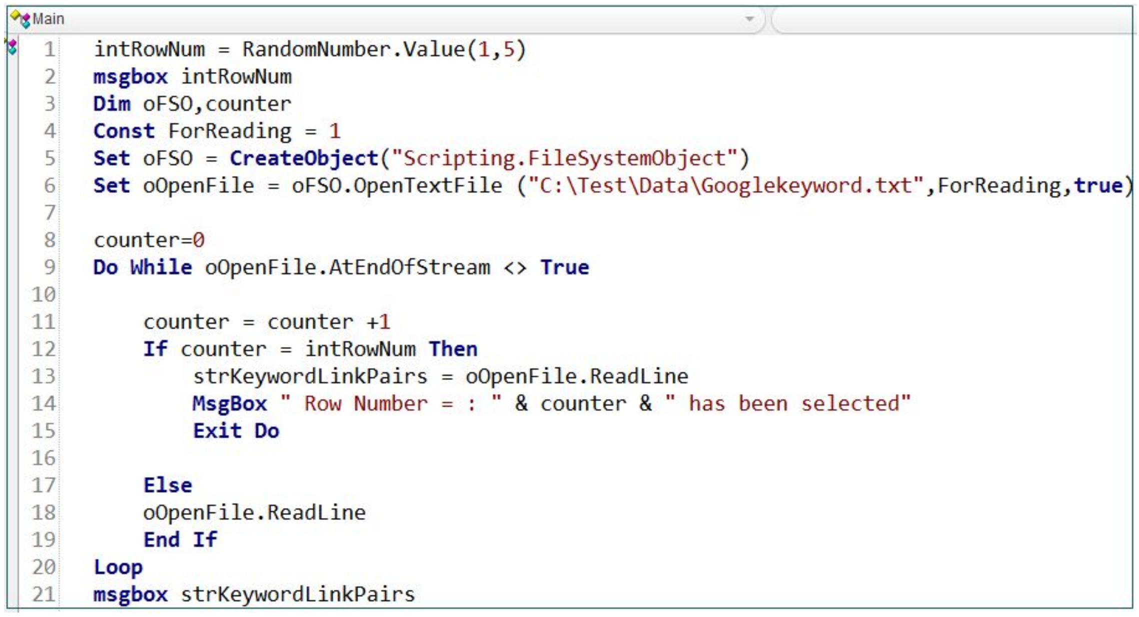
Step 5: Modify Test to select Random Number and Search Google

Let's first modify the text file by inserting five lines. Our aim to select the random row at run time and run our script accordingly.



```
UFT:News,QTP:Maps,Automation:Images,Email:Gmail
Indore:News,Jhansi:Maps,Newton:Images,Email:Gmail
UK:News,India:Maps,US:Images,Email:Gmail
Engineer:News,Doctor:Maps,Nurse:Images,Email:Gmail
SuperMan:News,Saktiman:Maps,Batman:Images,Email:Gmail
```

Navigate to UtilitiesUFT in UFT | Remove line 1 to 5 | Right Click Insert Step | Utilities Object | Random Number | provide 1, 5 in Value in vtItem row | Click Return value checkbox | Rename intRowNum | OK | Verify step intRowNum = RandomNumber.Value(1,5) in script | Modify the Script as following | Save | Run



```
 1  intRowNum = RandomNumber.Value(1,5)
 2  msgbox intRowNum
 3  Dim oFSO,counter
 4  Const ForReading = 1
 5  Set oFSO = CreateObject("Scripting.FileSystemObject")
 6  Set oOpenFile = oFSO.OpenTextFile ("C:\Test\Data\Googlekeyword.txt",ForReading,true)
 7
 8  counter=0
 9  Do While oOpenFile.AtEndOfStream <> True
10
11      counter = counter +1
12      If counter = intRowNum Then
13          strKeywordLinkPairs = oOpenFile.ReadLine
14          MsgBox " Row Number = : " & counter & " has been selected"
15          Exit Do
16
17      Else
18      oOpenFile.ReadLine
19      End If
20  Loop
21  msgbox strKeywordLinkPairs
```

Explain: In this example, we modified our script by removing lines 1 to 5.We introduced the code to generate random value from the range given in the RandomNumber function (Line-1). This random number will help to select the specific line in textfile(line-13). Now we have flexibility to create a file with multiple data entries and script logic to choose the specific value to drive the logic.

Step 6: Modify Test Setting at run time

This section is not related with step 1 to 5. In this section we are modifying test setting during run time..

Step 6A: Record script where UFT is unable to recorded object

Open UFT | New Test | Select GUITest "SettingMouseReply" | Google search "Hi" | Click Search Button | Click on | Search Help | Click language combo box English *| Select dansk | Click Mobilsøgning | Click language combo box* dansk *| Select English | Close IE | Save | Verify Script*

Explain: After performing the above steps, UFT unable to record the steps from the language drop-down. So, we need manually add these steps in script. We can do this by adding object in

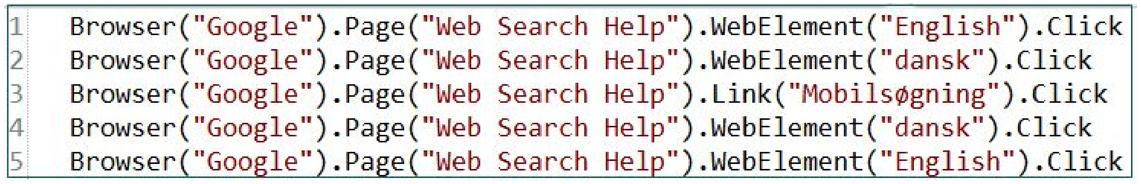
Step 6B: Add Object in OR from Object Spy

Open IE | Go to https://support.google.com/websearch/?hl=en | Navigate to SettingMouseReply in UFT | Tools | Object Spy | Click | Press CTRL key and drag Object Spy wizard to right hand side | release CTRL key | Click language combo box English *| Observer the hierarchy | Add Object in Object Repository | Navigate in IE and Select dansk in language combo* *box | Navigate UFT | Click | Click language combo box | Add Object in Object Repository | Click | Click on Mobilsøgning | Click | Close | Save*

Explain: This step add the little wired object that UFT was unable to record at first instance.

Step 6C: Modify Script to accommodate newly added Object

Navigate to SettingMouseReply in UFT | amend script with following lines after Link("Search help") step | Save | Run |



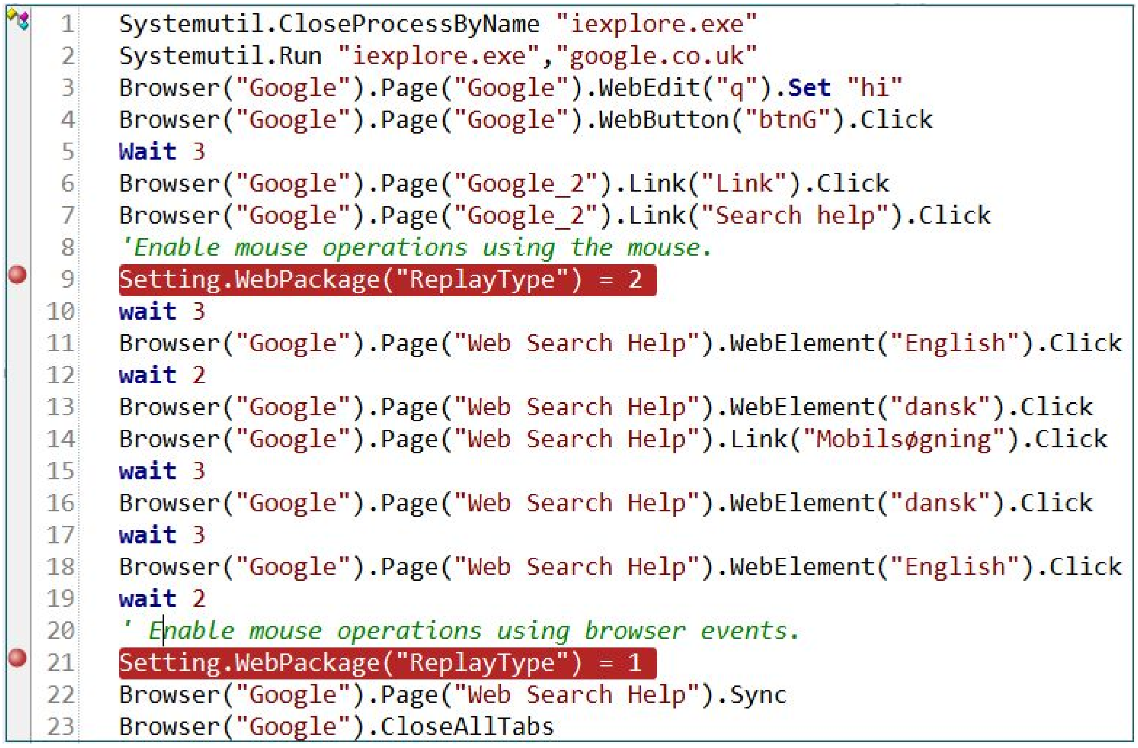
```
Browser("Google").Page("Web Search Help").WebElement("English").Click
Browser("Google").Page("Web Search Help").WebElement("dansk").Click
Browser("Google").Page("Web Search Help").Link("Mobilsøgning").Click
Browser("Google").Page("Web Search Help").WebElement("dansk").Click
Browser("Google").Page("Web Search Help").WebElement("English").Click
```

Explain: This script will fail. By default, the Click mouse method is managed as browser event in UFT. We need to change the reply setting and enable physical mouse to perform mouse operation of these tricky operation where UFT fail to recognize and proves incoherent to perform expected operations.

Warning: Your object name may be different. Please refer your object repository and provide correct hierarchy.

Step 6C: Modify Script to run mouse operation from mouse

Navigate to SettingMouseReply in UFT | insert Setting.WebPackage("ReplayType") = 2 line to enable mouse | insert Setting.WebPackage("ReplayType") = 1 to enable browser event for mouse operations | Save | Run



```
Systemutil.CloseProcessByName "iexplore.exe"
Systemutil.Run "iexplore.exe","google.co.uk"
Browser("Google").Page("Google").WebEdit("q").Set "hi"
Browser("Google").Page("Google").WebButton("btnG").Click
Wait 3
Browser("Google").Page("Google_2").Link("Link").Click
Browser("Google").Page("Google_2").Link("Search help").Click
'Enable mouse operations using the mouse.
Setting.WebPackage("ReplayType") = 2
wait 3
Browser("Google").Page("Web Search Help").WebElement("English").Click
wait 2
Browser("Google").Page("Web Search Help").WebElement("dansk").Click
Browser("Google").Page("Web Search Help").Link("Mobilsøgning").Click
wait 3
Browser("Google").Page("Web Search Help").WebElement("dansk").Click
wait 3
Browser("Google").Page("Web Search Help").WebElement("English").Click
wait 2
' Enable mouse operations using browser events.
Setting.WebPackage("ReplayType") = 1
Browser("Google").Page("Web Search Help").Sync
Browser("Google").CloseAllTabs
```

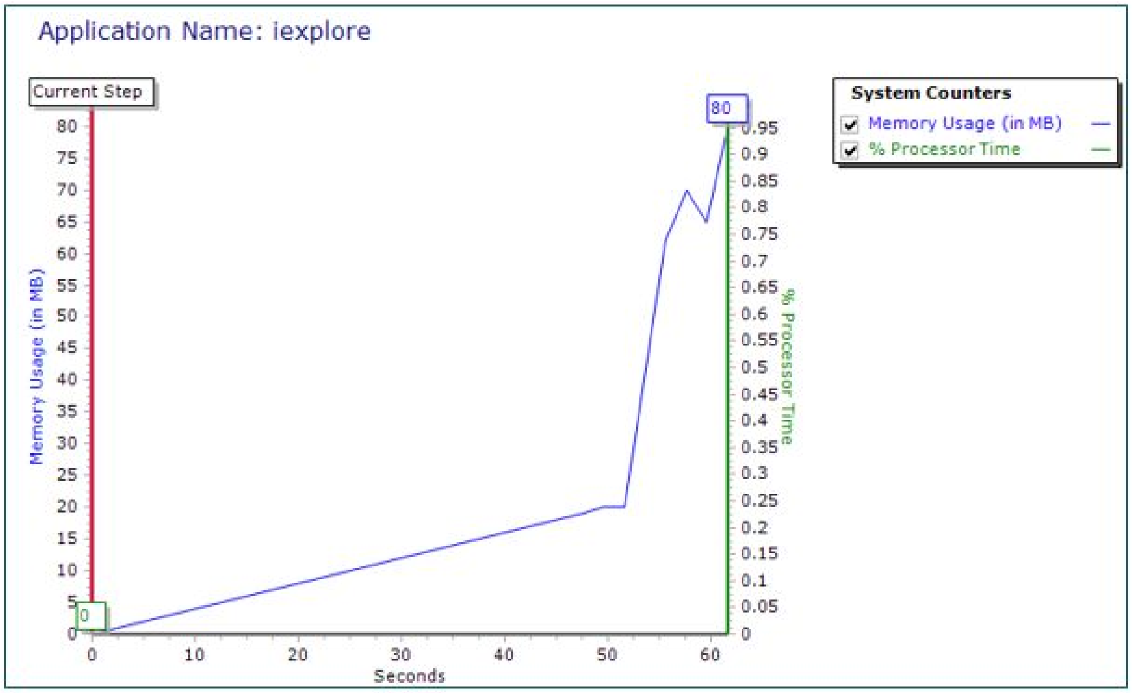
Explain: In this step, UFT set replaytype=2 i.e. mouse based action instead browser events. We need to reset back to the replaytype=1 after preforming action on tricky object.

7. Monitor System Performance

Navigate to "UtilitiesUFT" | File | Settings… | Local System Monitor | Check "Enable local system monitor every" | set 2 second | provide "iexplore" (not iexplore.exe) in Application to monitor | Select Memory Usage and %Processor Time in System counter |disable third counter row by clicking ☒ | Apply | OK | Run | View | Last Run Results | View | System Monitor | Close | Save | Close UFT

Warning: You may not get the system monitoring. You must close the UFT and re-open it. Also you need to provide application name without .exe extension e.g. "iexplore" instead of "iexplore.exe".

Open UFT | Navigate to "UtilitiesUFT" | Run | View | Last Run Results | View | System Monitor | Close | Save



Explain: UFT reporting can capture the system indicator during run time. There is a long list of counters listed in counter selection option related to disk, network, CPU and memory.

8 .Report should not show password given in script.

Open UFT | New |Test | Select GUITest with "GmailPasswordTest" | Record | Open IE | Navigate to gmail.com | provide user name "KrishanShukla" and "openSecrete " password | Close IE | Stop recording | modify password line as given below | insert wait wherever necessary | Save | Run

```
WebEdit("Passwd").Set Crypt.Encrypt("openSecrete ")
```

Explain: We need to share reports with various stakeholders and sometime we do not wish to reveal sensitive information in test e.g. salary, password, secret record etc. we can encrypt any string with help of Crypt.Encrypt(stringValue) that will reflect in UFT report. The following example shows that we can take value from somewhere (e.g. database) and encrypt it and pass in editbox. In this way, we will neither reveal it in script nor in UFT report.

```
strPwd= "Get_Password_from_Somewhere_Like_database"

en_pwd = Crypt.Encrypt(strPwd)

Browser("A").Dialog("B").WinEdit("bankpassword").SetSecure en_pwd
```