

## Topics to be covered

1. **Introduction** - Overview of automation and UFT(QTP)
2. Concepts of VBScript
3. Record and Playback, Object Repository, Actions
4. DataTables & Checkpoints
5. Synchronization, Smart Identification
6. Error handling and recovery scenarios
7. Environment Variables, library files, test results
8. Accessing databases
9. Working with XML
10. Descriptive programming
11. Framework and automation object Model.

### Q. Why we need test automation?

1. To save time and money.
2. Software tests have to be repeated often during development cycles to ensure quality. Every time source code is modified software tests should be repeated. For each release of the software it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming. Once created, automated tests can be run over and over again at no additional cost and they are much faster than manual tests.

### Q. What is regression testing?

Regression testing is testing existing software applications to make sure that a change or addition hasn't broken any existing functionality. Its purpose is to catch bugs that may have been accidentally introduced into a new build or release candidate, and to ensure that previously eradicated bugs continue to stay dead.

### Q. Why we need testing tools?

To perform our daily testing activities we need testing tools. Right from planning, requirements, creating build, test execution, defect logging.

### Various tools:

**Test Management Tool** – Plan tests, gather data from test run in different environments, log defects e.g HP QC

**Configuration management tools** - Configuration management is a technique or discipline to systematically manage, organize the changes in the documents, codes, artifacts and other entities during the development life cycle. E.g Perforce

**Test data Preparation Tools** - When an extensive range or volume of data is needed for testing then using the test data preparation tool is of great help. They are used for performance testing.

**Test Execution Tools:** Tools like Qtp are test execution tools which help execute your code.

**Performance testing tools:** These measure the performance of your application in peak traffic time e.g load runner.

**Incident Management Tools:** Unplanned interruption to an IT service is incident.

### **What is QTP or UFT?**

QTP Stands for quick test professional, owned by HP now named as Unified Functional Testing

### **Advantages**

- Developing automated tests using VBScript doesn't require a highly skilled coder and relatively easy when compared other object oriented programming languages.
- Easy to use, ease of navigation, results validation and Report generation.
- Readily Integrated with Test Management Tool(Hp-Quality Center) which enables easy scheduling and Monitoring.
- Can also be used for Mobile Application Testing.
- Since it is a Hp product, the full support is provided by HP and by its forums for addressing technical issues.

### **Disadvantages**

- Unlike Selenium, QTP works in Windows operating system only.
- Not all versions of Browsers are supported and the testers need to wait for the patch to be released for each one of the major versions.
- Having said that it is a commercial tool, the licensing cost is very high
- Even though scripting time is less, the execution time is relatively higher as it puts load on CPU & RAM.

### **Stages in automation**

**Feasibility Analysis:** Find out whether it is possible to automate given test.

**Appropriate Tool Selection:** Select a suitable tool.

**Evaluate a suitable framework:** Find out a suitable framework

**Build the Proof of Concept:** Proof of Concept (POC) is developed with an end to end scenario to evaluate if the tool can support the automation of the application.

## **Develop Automation Framework:**

### **Develop Test Script, Execute and Analyze:**

#### **UFT record and playback:**

Recording and playing back of recorded script:

#### **Significance of record and playback:**

- It is used as a preliminary method of investigation whether application is recognized by qtp and can be automated.
- Used to create a test a basic functionality of an application or feature that does not require long-term maintenance
- It can be used to record both mouse movement and keyboard inputs

#### **Modes of recording:**

**Normal Mode:** This is the default mode of recording; it records various objects and operations performed on application under test.

**Analog Mode:** It records mouse movements relative to screen in AUT together with keyboard movements.

**Low-Level Recording:** This records the exact co-ordinates of the objects independent of the fact whether UFT recognizes the object or NOT. It just records the co-ordinates, hence does NOT record mouse movements.

**Insight Recording:** UFT records operation based on its appearance and NOT based on its native properties.

#### **Q. How to select recording Modes?**

When you select recording button it gives you all the above options.

#### **Example of recording modes**

The Below Scenario is recorded in all the modes and see how the same action has been recorded under various circumstances.

Launch IE - <http://easycalculation.com/>

Click "Numbers" under "Algebra"

Click "Square Root" link

Enter a value to calculate the square root. Let us say 10

Hit Calculate

Script Recorded under Default, Analog and Low Level Recording Mode.

' **DEFAULT RECORDING MODE**

```
Browser("Free Online Math Calculator").Page("Free Online Math  
Calculator").Link("Numbers").Click
```

```
Browser("Free Online Math Calculator").Page("Numbers Calculator - Math").Link("Square  
Root").Click
```

```
Browser("Free Online Math Calculator").Page("Square Root  
Calculator").WebEdit("n").Set "10"
```

```
Browser("Free Online Math Calculator").Page("Square Root  
Calculator").WebButton("calculate").Click
```

```
' LOW LEVEL RECORDING MODE
```

```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Click  
235,395
```


```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Click  
509,391
```


```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Click  
780,631
```


```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Type "10"
```

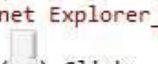
```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Click  
757,666
```

```
' INSIGHT RECORDING MODE
```


```
Browser("Free Online Math Calculator").InsightObject().Click
```


```
Browser("Free Online Math Calculator").InsightObject().Click
```

```
Browser("Free Online Math Calculator").InsightObject().Click
```

```
Browser("Free Online Math Calculator").InsightObject().Click
```

```
Window("Windows Internet Explorer").WinObject("Internet Explorer_Server").Type "10"
```

```
Browser("Free Online Math Calculator").InsightObject().Click
```

```
Browser("Free Online Math Calculator").InsightObject().Click
```

## Concepts Of Vbscript

### **Features OF VbScript:**

1. Lightweight scripting language, and has a lightning fast interpreter
2. VBScript, for the most part, is case insensitive. It has a very simple syntax, easy to learn and to implement.
3. Unlike C++ or Java, VBScript is an object-based scripting language and NOT an Object-Oriented Programming language
4. It uses Component Object Model (COM) in order to access the elements of the environment in which it is executing.
5. Successful execution of VBScript can happen only if it is executed in Host Environment such as Internet Explorer (IE), Internet Information Services (IIS) and Windows Scripting Host (WSH).

### **Variables in VbScript**

Dim is used to declare variable in vbscript for e.g dim avg. Here you need not declare the data type of variable by default all variables are of variant type.

- Variable Name must begin with an alphabet.
- Variable names cannot exceed 255 characters.
- Variables Should NOT contain a period(.)
- Variable Names should be unique in the declared context.
- The numeric values should be declared without double quotes.
- The String values should be enclosed within doublequotes("")
- Date and Time variables should be enclosed within hash symbol(#)

Variables have the following scope

- Dim
- Public
- and private

### **Constants**

```
const pi=3.14
```

```
Const myString = "VBScript"
```

```
Const myDate = #01/01/2050#
```

This will give an error as value of const cannot be changed

### **Operators**

Arithmetic operators: +, -, \*, /, %, ^ (exponent)

Comparison Operators: ==, <>, >, <,>=, <=

Logical Operators AND, OR, NOT, XOR

Concatenation Operators: +, &. + is an overloaded operator here when used with strings it does string concatenation and with numeric values it does addition.

## Decision making

If statement, if..else statement, if..elseif...else, Switch / select

```
Select Case Utilities.Uppercase(strMenuOption)
```

```
Case "HR SYSTEM OPTIONS"
```

```
Case "STORE PARAMETERS"
```

```
End select
```

## Loops

**For..Loop** – it executes the loop a given no of times

**For...each Loop** – for e.g

fruits is an array

```
fruits=Array("apple","orange","cherries")
```

```
Dim fruitnames
```

```
'iterating using For each loop.
```

```
For each item in fruits
```

```
fruitnames=fruitnames&item&vbnewline
```

```
Next
```

```
apple
```

```
orange
```

```
cherries
```

**While Wend...loop** -In a While..Wend loop, if the condition is True, all statements are executed until Wend keyword is encountered. For e.g

```
Dim Counter : Counter = 10
```

```
While Counter < 15 ' Test value of Counter.
```

```
Counter = Counter + 1 ' Increment Counter.
```

```
msgbox "Current values of counter is :"&Counter
```

```
Wend ' While loop exits if Counter Value becomes 15.
```

```
The Current Value of the Counter is : 11
```

```
The Current Value of the Counter is : 12
```

```
The Current Value of the Counter is : 13
```

```
The Current Value of the Counter is : 14
```

```
The Current Value of the Counter is : 15
```

**Do...While Loop** : The do..While statements will be executed as long as condition is True.(i.e.,) The Loop should be repeated till the condition is False. For e.g

```
Do While i < 5
```

```
    i = i + 1
```

```
    MsgBox "The value of i is : " & i
```

```
Loop
```

```
The value of i is: 1
```

```
The value of i is: 2
```

```
The value of i is: 3
```

```
The value of i is: 4
```

```
The value of i is: 5
```

**A Do..Until loop** is used when we want to repeat a set of statements as long as the condition is false. The Condition may be checked at the beginning of the loop or at the end of loop.

```
i=10
```

```
Do Until i>15 'Condition is False.Hence loop will be executed
```

```
    i = i + 1
```

```
    MsgBox "The value of i is:" & i
```

```
Loop
```

```
The value of i is : 11
```

```
The value of i is : 12
```

```
The value of i is : 13
```

```
The value of i is : 14
```

```
The value of i is : 15
```

```
The value of i is : 16
```

## Loop Control Statements

**Exit For statement:** Terminates the For loop statement and transfers execution to the statement immediately following the loop

```
Dim a : a=10
```

```
For i=0 to a Step 2 'i is the counter variable and it is incremented by 2
```

```
    MsgBox "The value is i is : " & i
```

```
        If i = 4 Then
```

```
            I = i*10 'This is executed only if i=4
```

```
            MsgBox "The value is i is : " & i
```

```
            Exit For 'Exited when i=4
```

```
        End If
```

```
Next
```

```
The value is i is : 0,
```

The value is i is : 2  
The value is i is : 4  
The value is i is : 40

**Exit Do statement:** Terminates the **Do While** statement and transfers execution to the statement immediately following the loop.

```
i = 0
Do While i <= 100
  If i > 10 Then
    Exit Do ' Loop Exits if i>10
  End If
  msgbox "The Value of i is : " &i
  i = i + 2
Loop
```

The Value of i is: 0  
The Value of i is : 2  
The Value of i is : 4  
The Value of i is : 6  
The Value of i is : 8  
The Value of i is : 10

## Number Conversion Functions:

Cdbl – A function which converts a given number of any variant subtype to double.  
Cint - A function which converts a given number of any variant subtype to Int.  
CLng - A Function, which converts a given number of any variant subtype to Long.  
CSng - A Function, which converts a given number of any variant subtype to Single.  
Hex - A Function which converts a given number of any variant subtype to Hexadecimal.  
FormatNumber - A Function, which would return an expression formatted as a number  
FormatPercent – A Function which would return an expression formatted as a percentage.

## Mathematical Functions:

Int - A Function, which returns the integer part of the given number.  
Fix - A Function, which returns the integer part of the given number.  
Log - A Function, which returns the natural logarithm of the given number. Negative numbers disallowed.  
Oct - A Function, which returns the Octal value of the given percentage.  
Hex - A Function, which returns the Hexadecimal value of the given number.  
Rnd - A Function, which returns a random number between 0 and 1.  
Sgn - A Function, which returns a number corresponding to the sign of the specified number.



Sqr - A Function, which returns the square root of the given number. Negative numbers disallowed.

Abs - A Function, which returns the absolute value of the given number.

Exp - A Function, which returns the value of e raised to the specified number.

Sin - A Function, which returns sine value of the given number.

Cos - A Function, which returns cosine value of the given number.

Tan - A Function, which returns tan value of the given number.

## Strings:

InStr - The InStr Function returns the first occurrence of one string within another string. The search happens from left to right.

`InStr([start,]string1,string2[,compare])`

- Compare - 0 = vbBinaryCompare - Performs Binary Comparison(Default)
- 1 = vbTextCompare - Performs Text Comparison

InstrRev - Returns the first occurrence of the specified substring. Search happens from Right to Left

Lcase - Returns the lower case of the specified string.

Ucase - Returns the Upper case of the specified string.

Left - The Left Function returns a specified number of characters from the left side of the given input string. For e.g

For e.g `Left(String, Length)`

```
var="Microsoft VBScript"
```

```
Msgbox Left(var,2)
```

Ans Mi

Right - The Right Function returns a specified number of characters from the Right side of the given input string. For e.g

For e.g `Right(String, Length)`

```
var="Microsoft VBScript"
```

```
Msgbox Right(var,2)
```

Ans pt

LTrim - The Ltrim Function removes the blank spaces that are there on the left side of the string.

RTrim - The Rtrim Function removes the blank spaces that are there on the Right side of the string.

Trim - The trim Function removes the blank spaces that are there on the both sides of the string.

Len - Calculates len of string. For e.g `len("The") = 3`

Mid - The Mid Function returns a specified number of characters from a given input string.

```
Mid(String,start[,Length])  
var="Microsoft VBScript"  
Msgbox Mid(var,2,5)  
Ans "icros"
```

Replace - The Replace Function replaces a specified part of a string with a specific string a specified number of times.

**Syntax** - Replace(string, find, replacewith, start, count, compare)

**string**, a Required Parameter. The Input String from to be searched for replacing.

**find**, a Required Parameter. The Part of the String that will be replaced.

**replacewith**, a Required Parameter. The replacement string, which would be replaced against the find parameter.

**start**, an Optional Parameter. Specifies the start position from where the string has to be searched and replaced. Default value is 1.

**count**, an Optional Parameter. Specifies the number of times the replacement has to be performed.

**compare**, an Optional Parameter. Specifies the comparison method to be used. Default value is 0.

0 = vbBinaryCompare - Performs a binary comparison

1 = vbTextCompare - Performs a Textual comparison

```
var="This is VBScript Programming"  
"is' replaced by ##"  
Msgbox Replace(var,"is", "##")  
Ans - Th## ## VBScript Programming
```

"s' is replaced by ## for the next 2 occurrences.

```
MSGBOX Replace(var,"s", "##",1,2) & "<br />")
```

Ans Thi## i## VBScript Programming

**Space** - Fills a string with the specified number of spaces.

**StrComp** -The StrComp Function returns an integer value after comparing the two given strings. It can return any of the three values -1, 0 or 1 based on the input strings to be compared.

```
StrComp(string1,string2,compare)
```

If String 1 < String 2 then StrComp returns -1

If String 1 = String 2 then StrComp returns 0

If String 1 > String 2 then StrComp returns 1

```
MSGBOX StrComp("Microsoft","Microsoft")
MSGBOX StrComp("Microsoft","MICROSOFT")
MSGBOX StrComp("Microsoft","MiCrOsOfT")
MSGBOX StrComp("Microsoft","MiCrOsOfT",1)
MSGBOX StrComp("Microsoft","MiCrOsOfT",0)
```

Line 1 :0

Line 2 :1

Line 3 :1

Line 4 :0

Line 5 :1

**String** - The String Function fills a string with the specified character the specified number of times.

**String(number,character)**

**Number**, a Required Parameter. An integer value, which would be repeated for the specified number of times against the character parameter.

**Character**, a Required Parameter. Character value, which has to be repeated for the specified number of times.

```
MSGBOX String(3,"$")
$$$
```

**StrReverse** - Returns a String after reversing the sequence of the characters of the given string.

StrReverse(string)

## ARRAYS:

'Method 1 : Using Dim

Dim arr1() 'Without Size

'Method 2 : Mentioning the Size

Dim arr2(5) 'Declared with size of 5

'Method 3 : using 'Array' Parameter

Dim arr3

arr3 = Array("apple","Orange","Grapes")

Assigning Values to an array

```
Dim arr(5)
arr(0) = "1"           'Number as String
arr(1) = "VBScript"   'String
arr(2) = 100           'Number
arr(3) = 2.45          'Decimal Number
arr(4) = #10/07/2013# 'Date
arr(5) = #12.45 PM#   'Time
```

### **Multi Dimension Arrays:**

```
Dim arr(2,3)           ' Which has 3 rows and 4 columns
arr(0,0) = "Apple"
arr(0,1) = "Orange"
arr(0,2) = "Grapes"
arr(0,3) = "pineapple"
arr(1,0) = "cucumber"
arr(1,1) = "beans"
arr(1,2) = "carrot"
arr(1,3) = "tomato"
arr(2,0) = "potato"
arr(2,1) = "sandwich"
arr(2,2) = "coffee"
arr(2,3) = "nuts"
```

### **Redim Statement:**

ReDim Statement is used to Declare dynamic-array variables and allocate or reallocate storage space.

```
ReDim [Preserve] varname(subscripts) [, varname(subscripts)]
```

Preserve - An Optional parameter used to preserve the data in an existing array when you change the size of the last dimension.

varname - A Required parameter, which denotes Name of the variable, which should follow the standard variable naming conventions.

subscripts - A Required parameter, which indicates the size of the array.

```

Dim a()
i=0
redim a(5)
a(0)="XYZ"
a(1)=41.25
a(2)=22

REDIM PRESERVE a(7)
For i=3 to 7
a(i)= i
Next

'to Fetch the output
For i=0 to ubound(a)
Msgbox a(i)
Next
XYZ
41.25
22
3
4
5
6
7

```

**LBound** A Function, which returns an integer that corresponds to the smallest subscript of the given arrays.

**UBound** A Function, which returns an integer that corresponds to the Largest subscript of the given arrays.

**Split** A Function, which returns an array that contains a specified number of values. Splitted based on a Delimiter.

**Join** A Function, which returns a String that contains a specified number of substrings in an array. This is an exact opposite function of Split Method.

**Filter** A Function, which returns a zero based array that contains a subset of a string array based on a specific filter criteria.

**IsArray** A Function, which returns a boolean value that indicates whether or not the input variable is an array.

**Erase** A Function, which recovers the allocated memory for the array variables.

## Date Functions

**Date** A Function, which returns the current system date

CDate A Function, which converts a given input to Date

DateAdd A Function, which returns a date to which a specified time interval has been added

DateDiff A Function, which returns the difference between two time period

DatePart A Function, which returns a specified part of the given input date value

DateSerial A Function, which returns a valid date for the given year, month and date

FormatDateTime A Function, which formats the date based on the supplied parameters

IsDate A Function, which returns a Boolean Value whether or not the supplied parameter is a date

Day A Function, which returns an integer between 1 and 31 that represents the day of the specified Date

MonthA Function, which returns an integer between 1 and 12 that represents the month of the specified Date

Year A Function, which returns an integer that represents the year of the specified Date

MonthName A Function, which returns Name of the particular month for the specified date

WeekDay A Function, which returns an integer(1 to 7) that represents the day of the week for the specified day.

WeekDayName A Function, which returns the weekday name for the specified day.

#### Time Functions

Function	Description
----------	-------------

Now	A Function, which returns the current system date and Time
-----	--

Hour	A Function, which returns and integer between 0 and 23 that represents the Hour part of the the given time
------	--

Minute A Function, which returns an integer between 0 and 59 that represents the Minutes part of the given time

Second A Function, which returns an integer between 0 and 59 that represents the Seconds part of the given time

Time A Function, which returns the current system time

Timer A Function, which returns the number of seconds and milliseconds since 12:00 AM

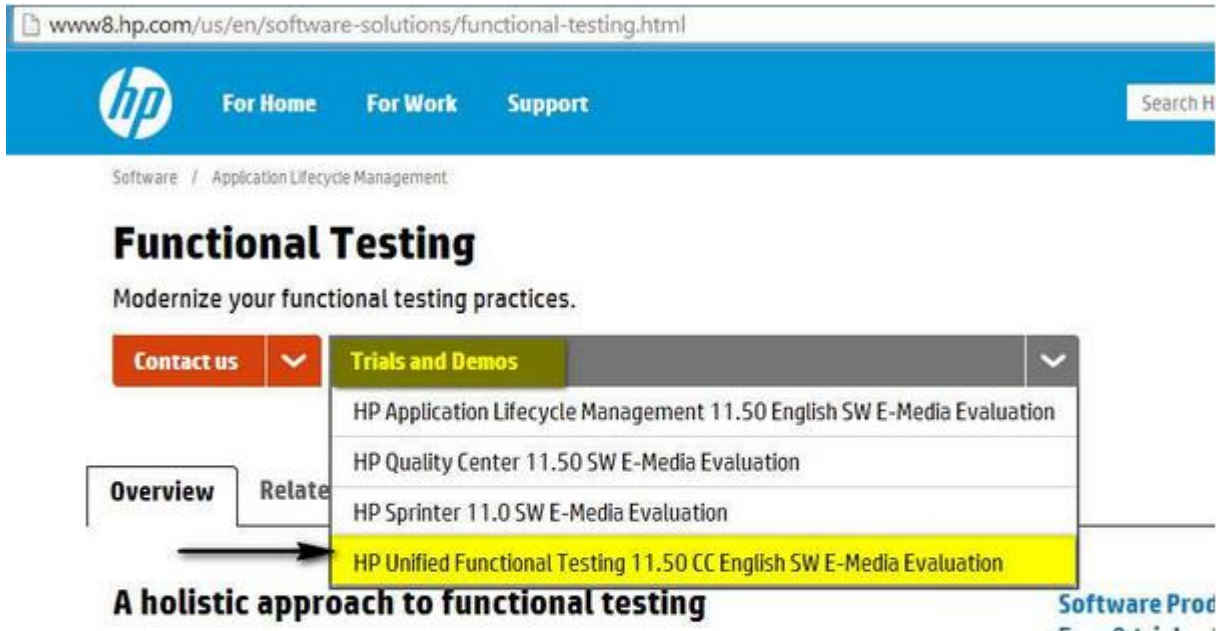
TimeSerial A Function, which returns the time for the specific input of hour, minute and second

TimeValue A Function, which converts the input string to a time format

## Installing QTP / UFT

Goto Hp Website Register / sign in.

Click "Trials and Demos" link and select "Hp Unified Functional Testing 11.50 CC English SW E-Media Evaluation" as shown below:



The screenshot shows the HP website's 'Functional Testing' page. The URL is [www8.hp.com/us/en/software-solutions/functional-testing.html](http://www8.hp.com/us/en/software-solutions/functional-testing.html). The page features the HP logo and navigation links for 'For Home', 'For Work', and 'Support'. A search bar is visible on the right. Below the navigation, the text 'Software / Application Lifecycle Management' is displayed. The main heading is 'Functional Testing' with the subtext 'Modernize your functional testing practices.' A dropdown menu is open under 'Trials and Demos', listing several options. The option 'HP Unified Functional Testing 11.50 CC English SW E-Media Evaluation' is highlighted in yellow. An arrow points from the 'Overview' tab to this highlighted option. Below the dropdown, the text 'A holistic approach to functional testing' is visible, along with a partially visible 'Software Prod' link.

**Step 2** - Upon Selecting "Hp Unified Functional Testing 11.50", the download wizard opens. Fill in the Personal details and click next



The screenshot shows the HP download wizard progress bar. It consists of three steps: '1 About yourself', '2 Terms of service', and '3 Download'. The '2 Terms of service' step is currently active. Below the progress bar, the text 'HP Unified Functional Testing 11.50 CC English SW E-Media Evaluation' is displayed. A close button (X) is visible on the right side of the progress bar.

Read the terms and conditions, Click next and start downloading



## HP Unified Functional Testing 11.50 CC English SW E-Media Evaluation

**Please review this first:**

Letter, HP UFT 11.50 English

PDF 0.08 MB

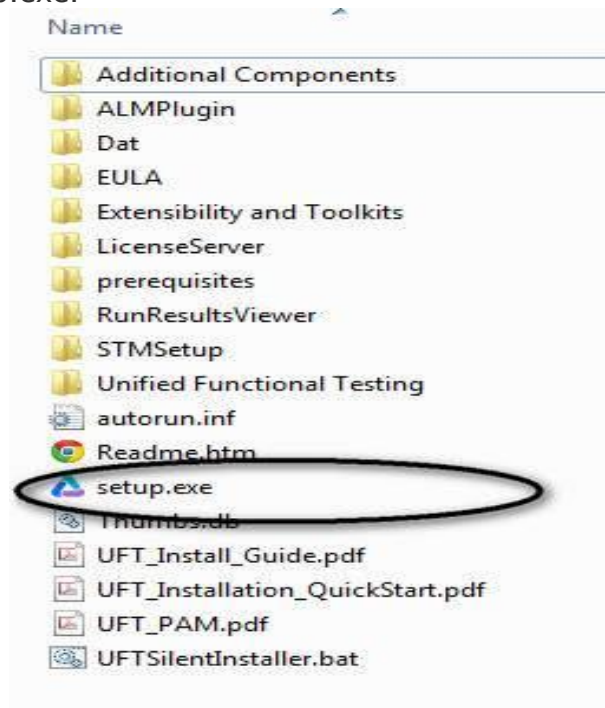
**You may now download your software and supporting materials.**

For License keys or to download content again, please refer to our download confirmation email sent to [tshanmuganathan@gmail.com](mailto:tshanmuganathan@gmail.com). Review our [FAQ](#) for more helpful information.

Name	File Size	Download Methods	
		Using HP Download Manager	?
Software, HP UFT 11.50 English ( T6510-15 080.zip )	1754 MB	Download	
Letter, HP UFT 11.50 English	0.08 MB	Download	

The downloaded file will be in .RAR format

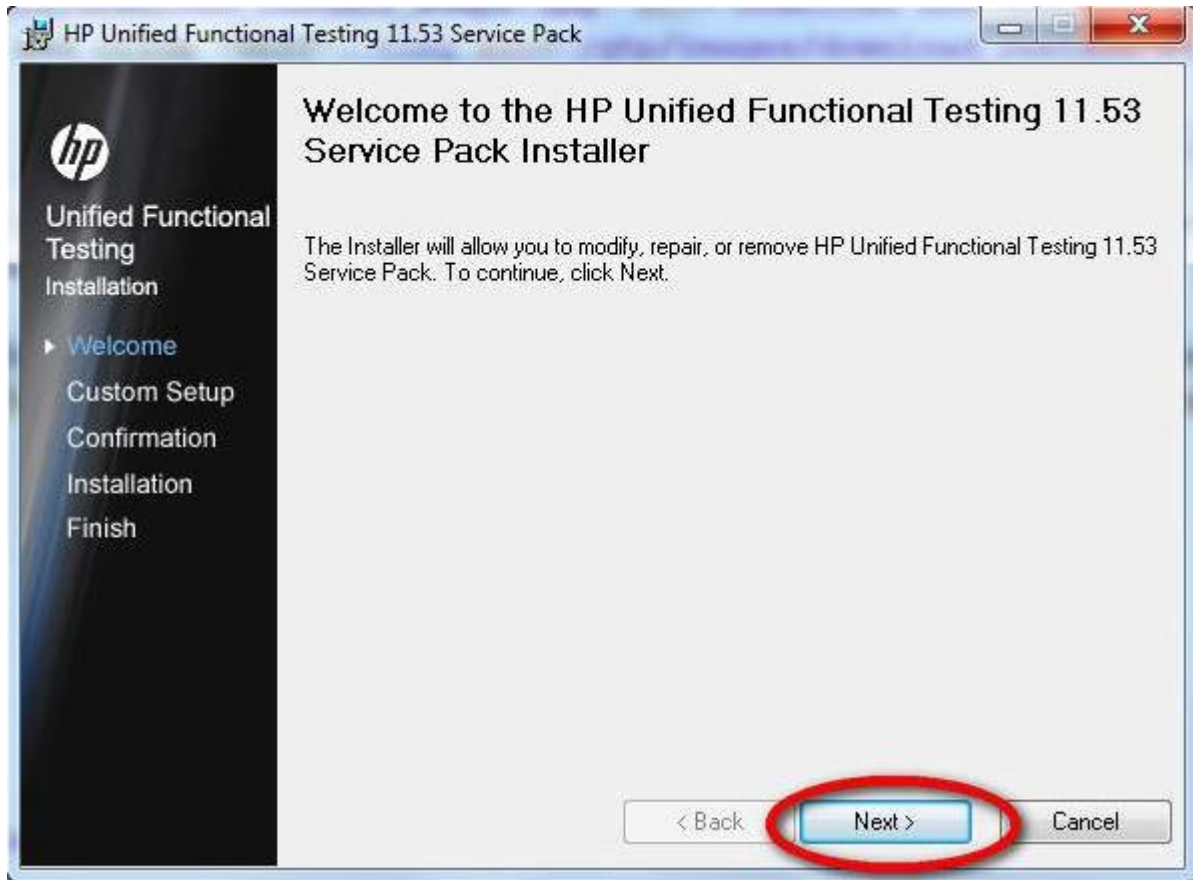
Now you need to unzip the archive and the folder contents would be as shown below and execute the Setup.exe.



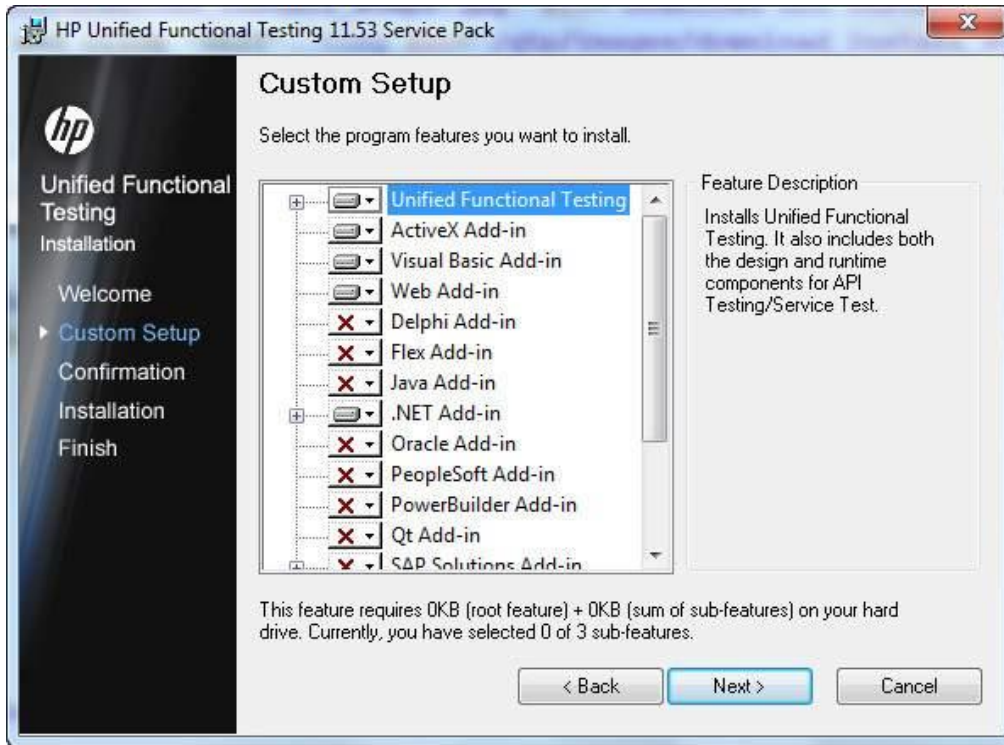
Upon Executing the Setup File, in order to install, select "Unified Functional Testing Set up" from the list as shown below:



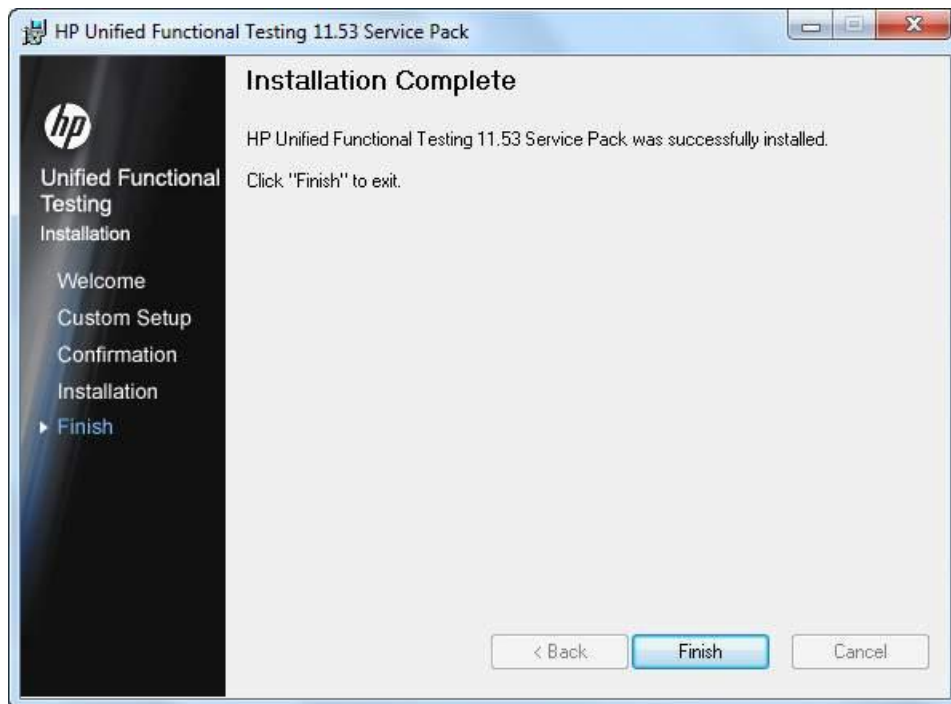
Click next to continue



In the Custom Set up Window, select the plugins that are required for your automation. i.e. You Should select the plugins based on the technology of your application under test. For Example, If your application is based on .NET then you should ensure that you select .NET.

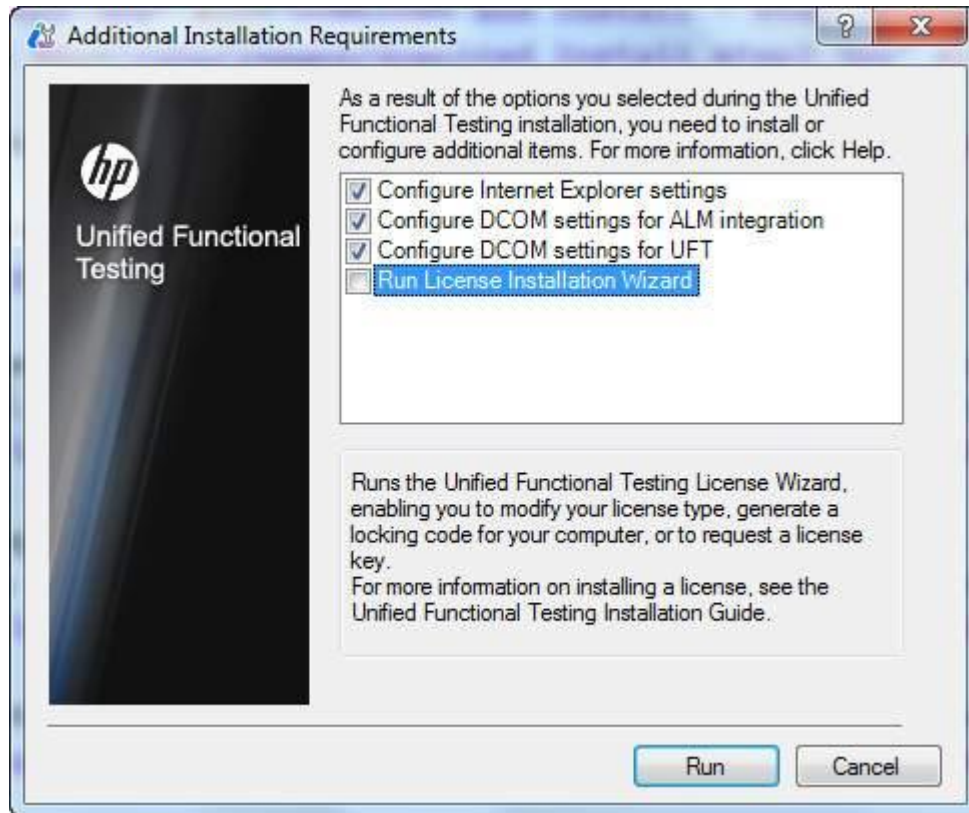


Upon Selecting the required plugins for Installation, Click Next and upon completion of the installation you will end up with a Finish button Window.

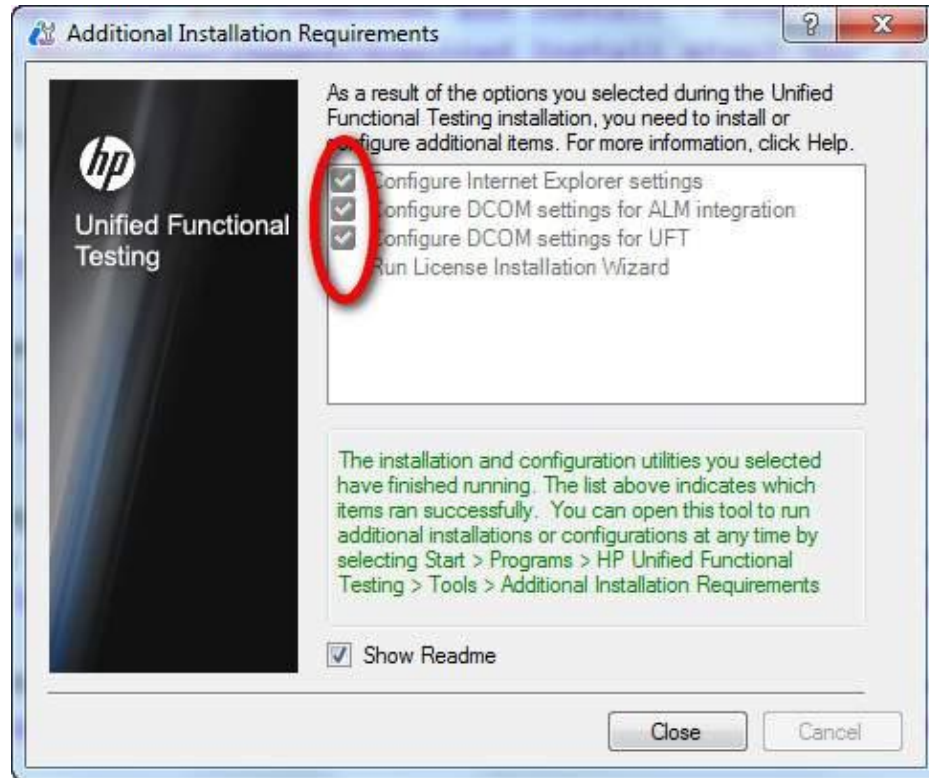


Once you complete your installation, the "Additional Installation Requirements" Dialog box opens. Select everything in the list other than "Run License Installation Wizard"

and click "RUN". We Need NOT select "Run License Installation Wizard" because we are installing the trial version which by default gives a license for 30 days.

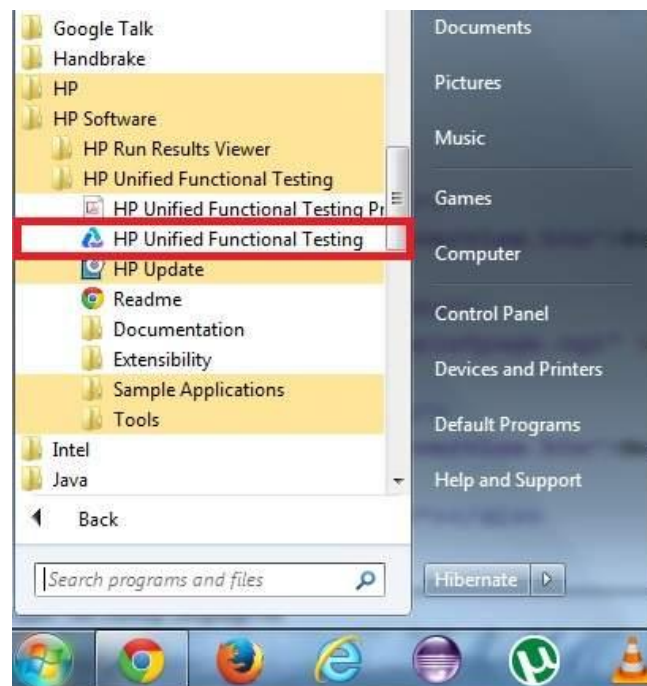


Upon completion of Additional Installation Requirements, a tick mark is shown which inturn states that the components are installed successfully. Now, click close:



### Launching UFT / QTP Addins Page

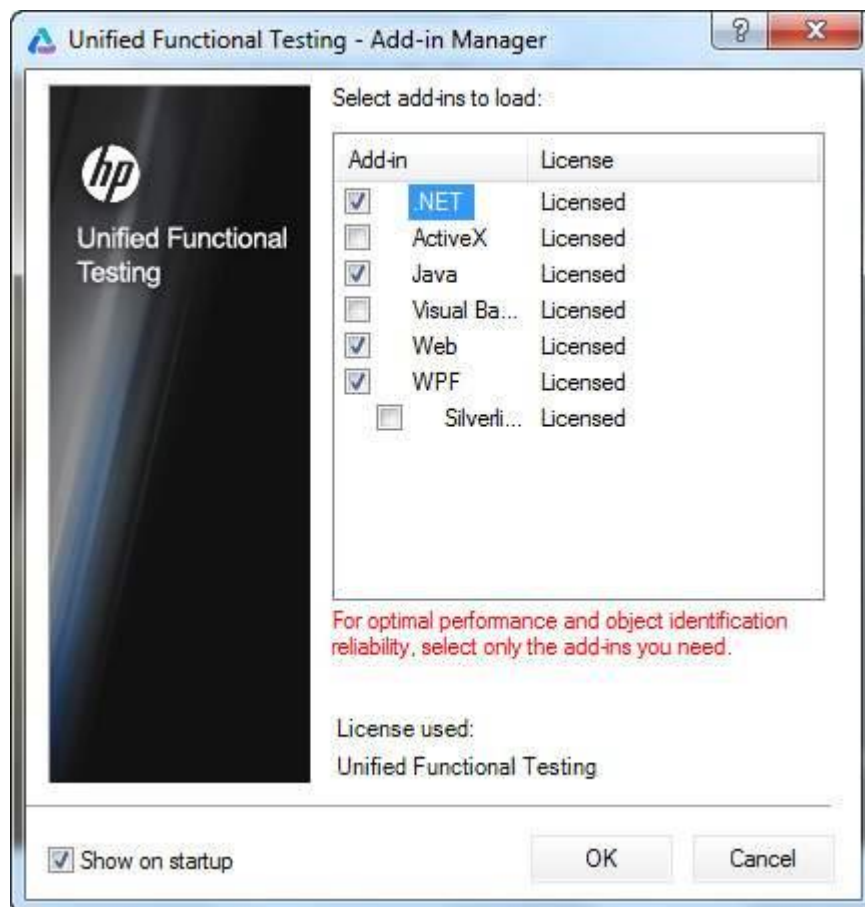
After Installation, application can be launched from the Start Menu as shown in



The License page appears. You can click on continue as we have installed the trial license



The Addins Dialog box opens for the user to select the required addins' DONOT load all the addin's but just the required addins and click "Ok" button

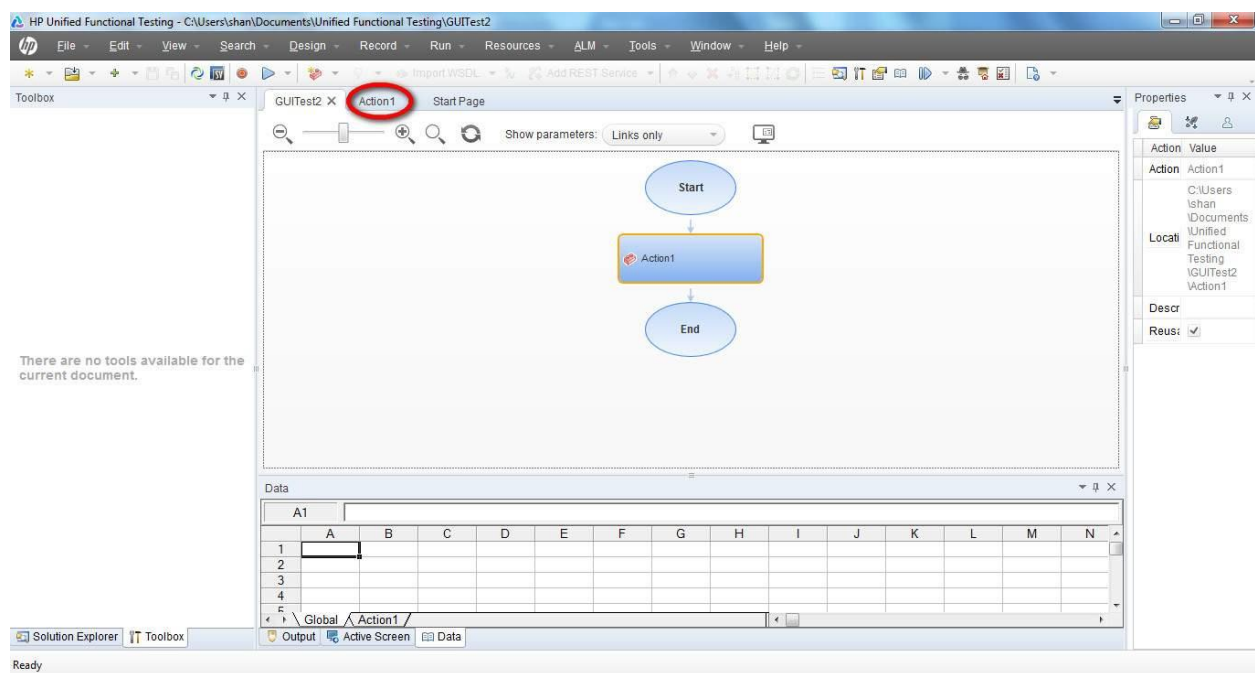


## QTP Record And playback

Recording a test corresponds to recording the user actions of the application under test so that UFT automatically generates the scripts that can be played back. Record and Playback can give us the first impression if the tool can support the technology or NOT if the initial settings are done correctly.

Click on "New" test from the Start Page as shown below:

Upon Clicking, "New" Link, the new test window opens and the user need to select the test type. Select "GUI Test", give a name for the test and also the location where it needs to be saved.

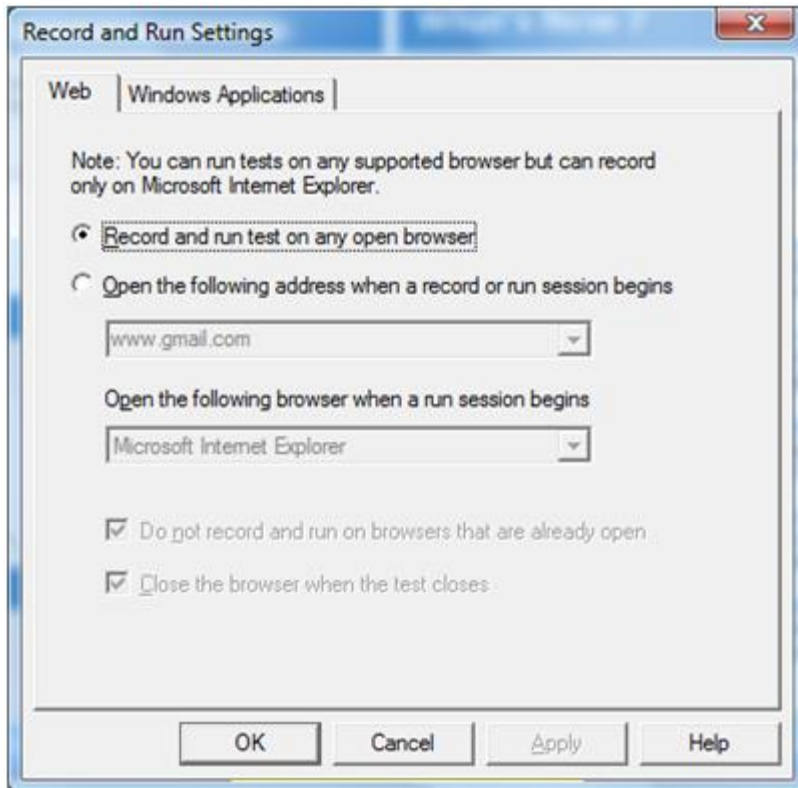


As the new test is created click on Action1

Now click on Record button and select record and run settings, the record and run settings dialog appears. Based on the type of application, one can select i.e Web, Java, Windows Applications.

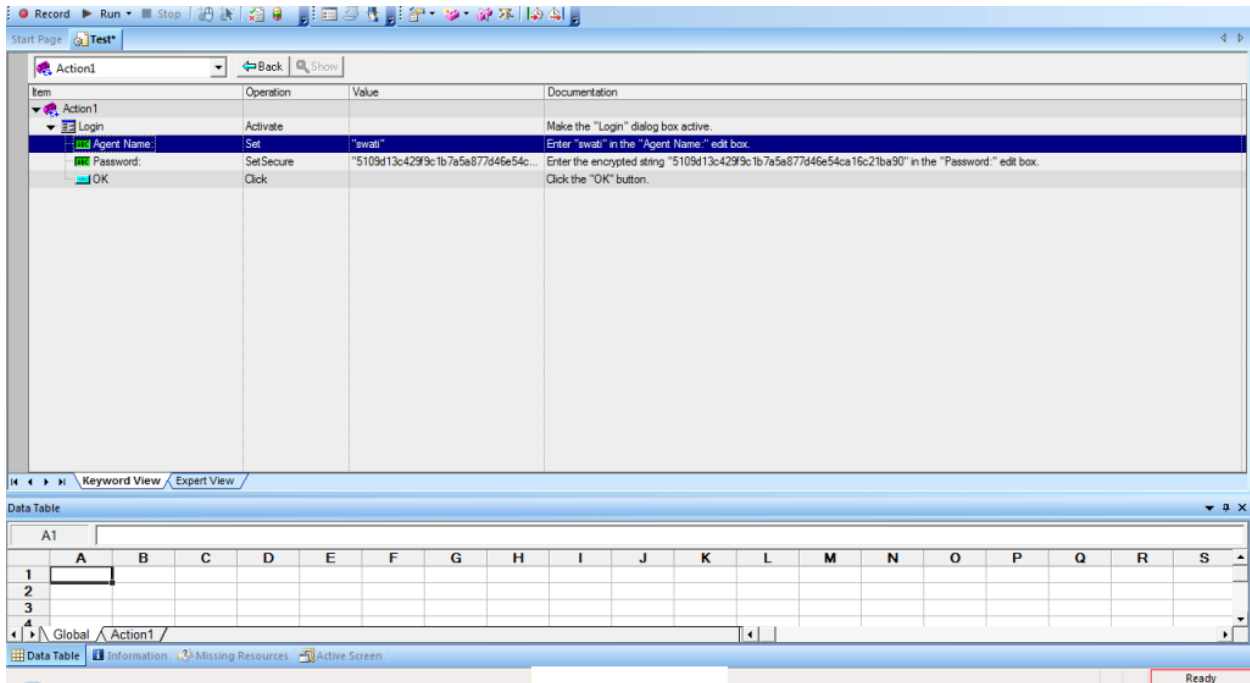
**Note:** - The no of addins you have installed that many no of tabs will appear in record and play back window.



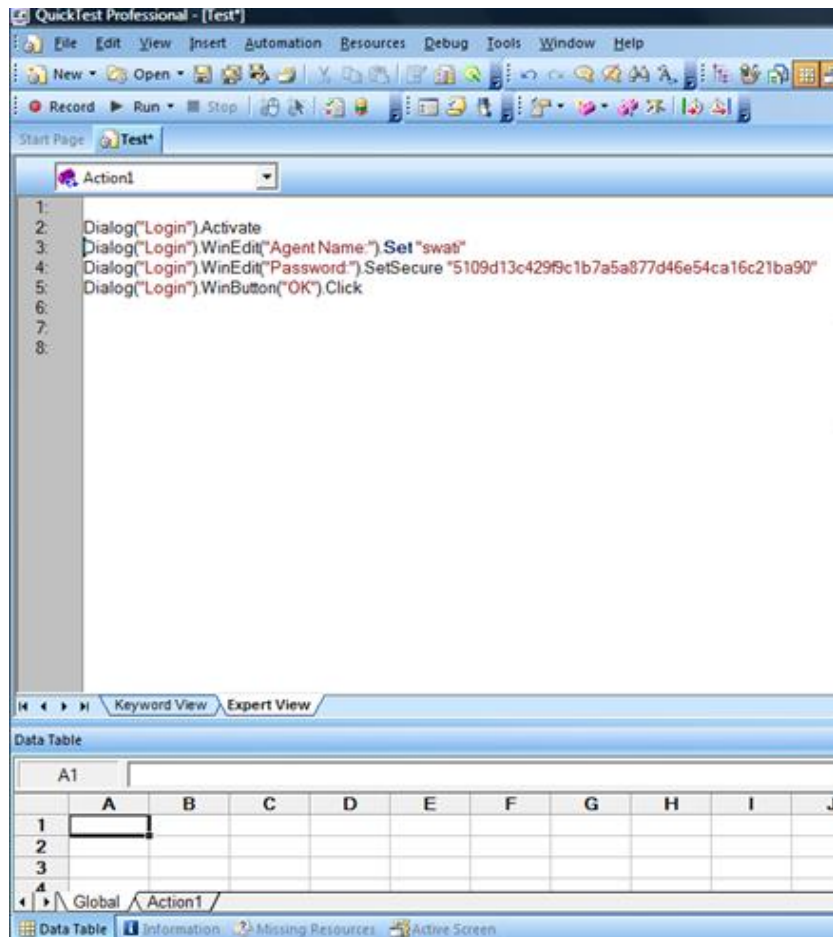


For e.g we will take flight application which comes with QTP and enter agent name and password on the login screen.

The following screen shows the keyword view. As you can see the actions are in form of a table with columns: Item, operation, value and documentation.



The corresponding keyword view for the above test will look like.

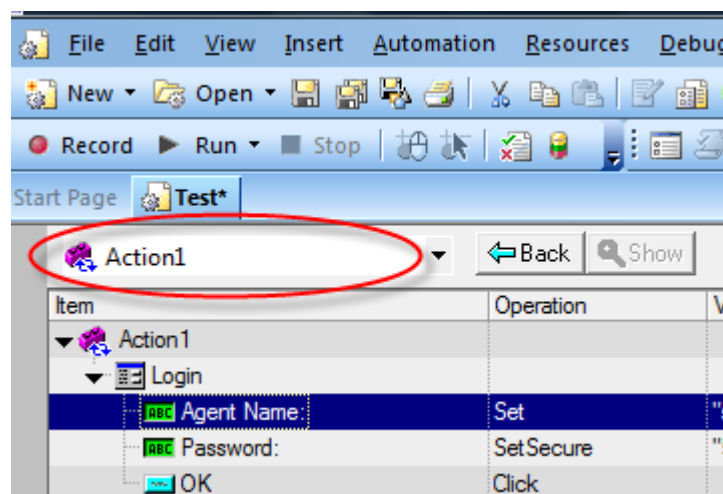


## Actions

All tests in QTP are basically composed of actions. Each actions is nothing but a sequence of statements performing a sequence of actions (activities). Actions make your test look modular.

For e.g. if someone has to send a mail, he will login into his Gmail account, compose mail, send and logout. These can be three actions although they can be written in one action.

If they all are written together the lines of code will be large and difficult to mainatian, hence it is better to divide it into actions. The test is essentially call to action 1, call to action 2, call to action 3 and so on.



In the above dropdown you can click and view all the actions associated with the test Here we can also select and go to that particular action, view and edit.

There is also a concept of **re-usable actions**, these actions can be reused multiple times within the test they were created or by other tests also. By default each action is reusable although that can be changed if needed.

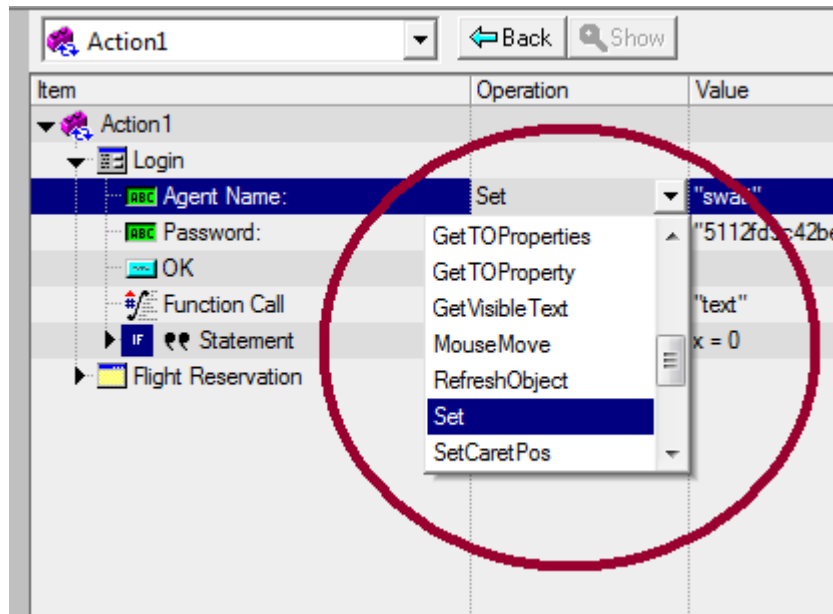
**Object Hierarchy-** Qtp uses a tree hierarchy to store and recognize objects. In our case agent name and password are child objects to the container object login which is a dialog. Container objects are window, dialog in windows environment and browser and page in web environment.

## Explanation of keyword view

**Item:** Each step is a sequence of events performed on an item, item can be one of the following:

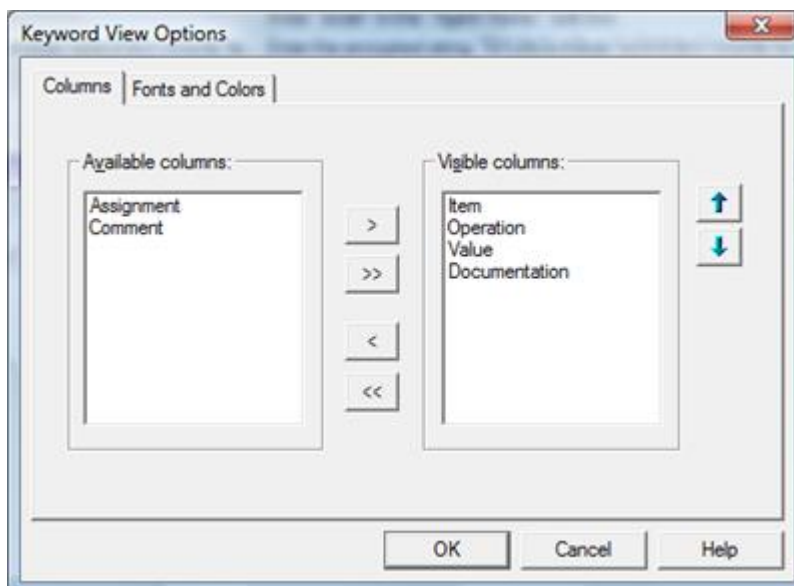
- Test Object e.g Agent name
- Utility object like datatable object
- Statements like loops
- Function call

**Operation:** It shows the operation that is performed on the item. When this column is clicked for a particular item it lists all the available operations that can be performed on the object.

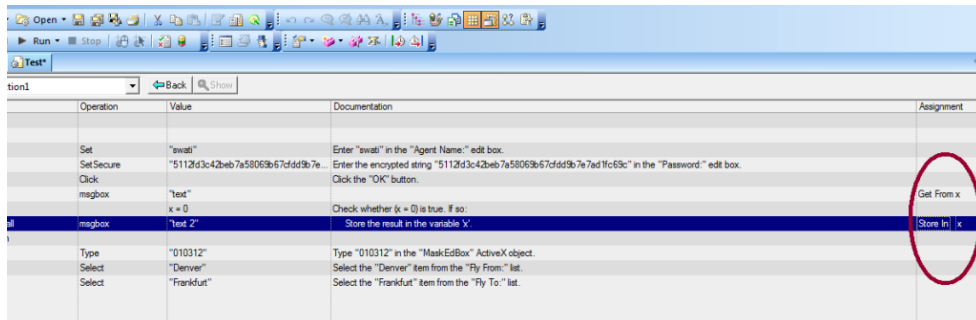


**Value:** This column can be considered as the argument of a statement displayed. In our example, the message box function call has a value "Text". There can be multiple arguments to one statement in which case this column will be divided accordingly.

**Documentation:** A read only column that translates a statement into easily understandable comments. There are two other columns Assignment and Comment that can be added to the Keyword view table. To do so, the option to select is **Tools->View> Options** and choose the required columns from the list displayed.



**Assignment:** This column is not very widely used and what this does is assigns a value to or gets a value from a variable. [click on image to enlarge]



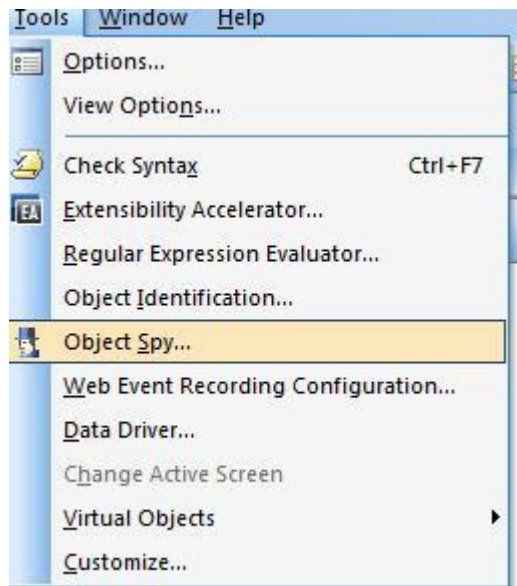
SoftwareTestingHelp.com

**Comments:** The tester can write anything under this column. It will be treated as comments in the expert view.

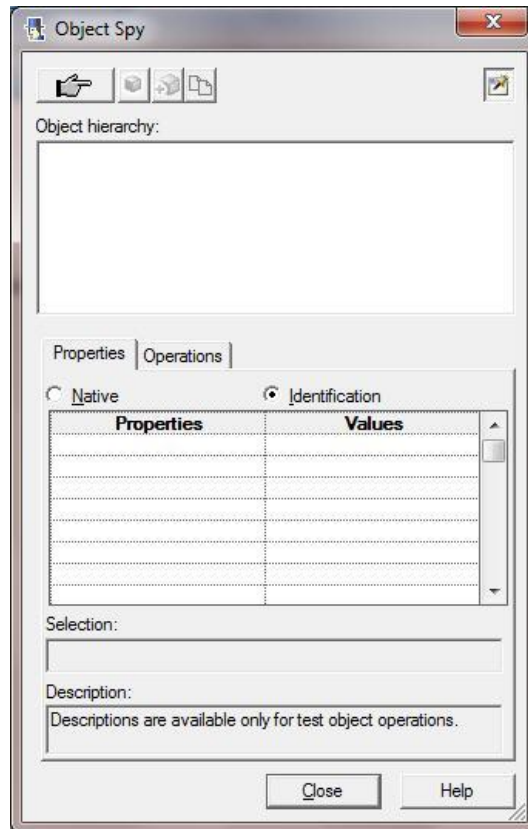
### Obect Repository:



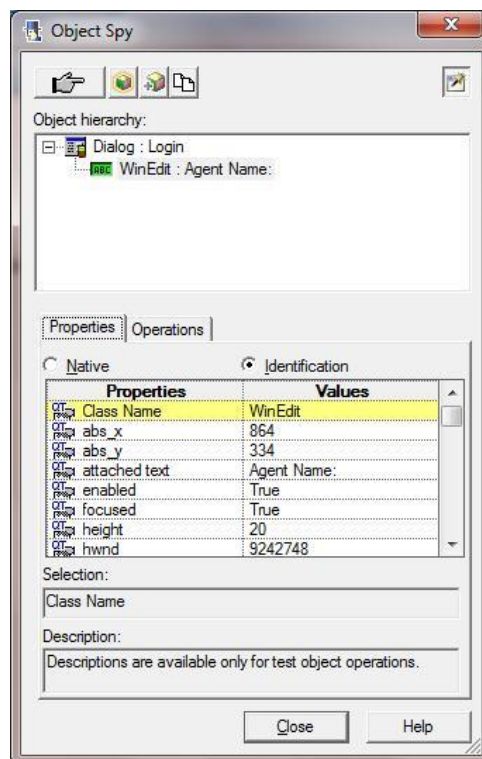
Click the icon or click Tools -> Object Spy to access the object repository



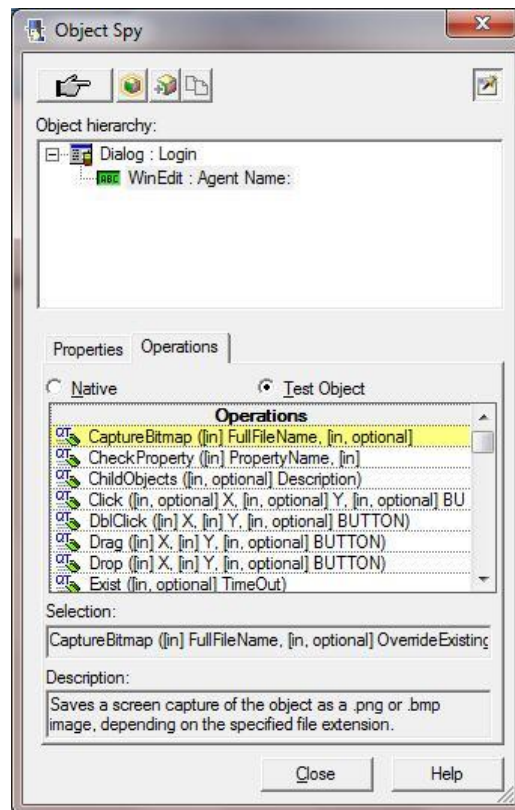
When Object Spy is launched, this is how its interface looks:




By clicking on the hand icon you can point to any object you wish to examine.



As you can see, it shows all the properties of the object and the corresponding values. It shows the object hierarchy. There is also an 'Operations' tab which when clicked displays all the operations that can be performed on the test object:



The spy not only displays the properties, it also has a provision that lets the users add a certain object to the OR. You can do that by clicking on the OR  icon with a + in it.

## **Object Repository**

It stores the set of properties that uniquely identifies the Object (description) and also names the object for the sake of identification in our test, based on its most prominent feature.

The objects that are acted upon during record process get automatically added to the OR. You can also use Ctrl + R to invoke Object repository.

There are two kinds of Object repositories

- a. Shared repository
- b. Local repository

Each action has a separate local repository. To make an OR as a shared repository we need to use **Object Repository Manager**.

Also, a shared repository is read-only in the action level and any changes to it have to be made via 'Object Repository manager.

Even though an action may not use a local OR as it may be using a shared repository it will have a local repository.

If a local as well as shared repository has an object with same name the test will use a local object first.

There can be more than one Shared OR's associated to the same action. If Shared OR1 and Shared OR2 have one object named OBJ1 each and if the action calls for OBJ1 then the order in which the shared ORs were associated will be considered. That means, if Shared OR1 was first associated then the OBJ1 from Shared OR1 will be taken into account.

## **Checkpoints**

When the user wants to insert a checkpoint, one has to ensure that most of the checkpoints are supported during the recording sessions only. Once the user stops recording, checkpoints are NOT enabled.

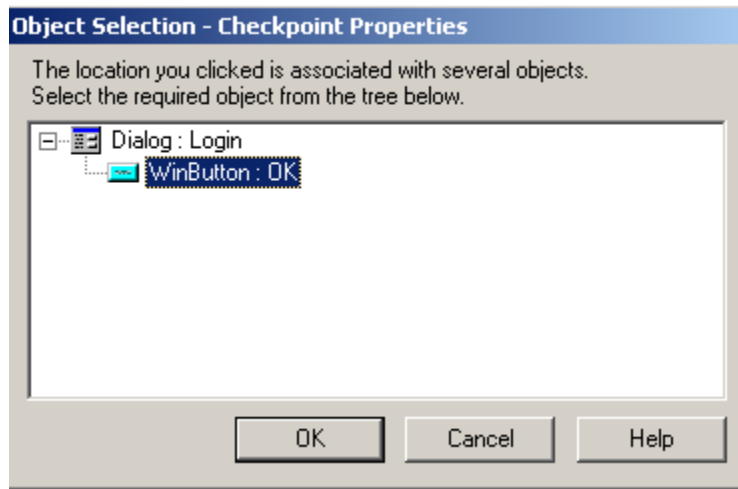
Below is the checkpoint menu, when the user is NOT in the recording mode.

### **Standard Checkpoint**

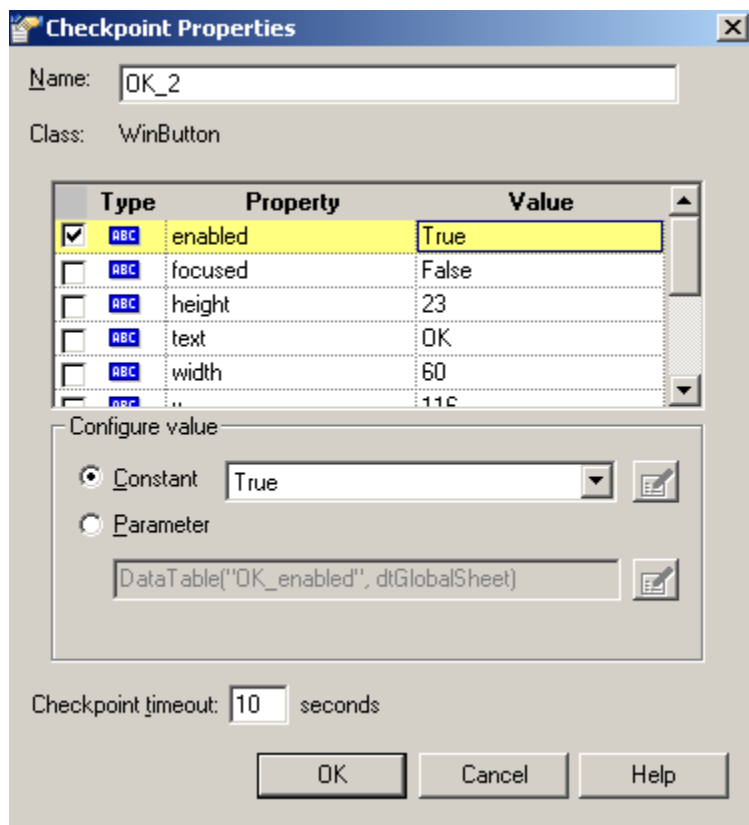
It is used to check object property value and compares the actual value with the expected value during run time. It can be set during recording or editing a test. Follow the below steps for inserting a standard checkpoint.

1. Start recording, launch flight supplication
2. Enter agent name and password in respective textboxes.
3. Go to Insert > Checkpoint > Standard Checkpoint
4. Point to the ok button on the login screen of flight application, mouse arrow turns into hand select the ok button
5. Object selection properties dialog box appears





Click on OK button.



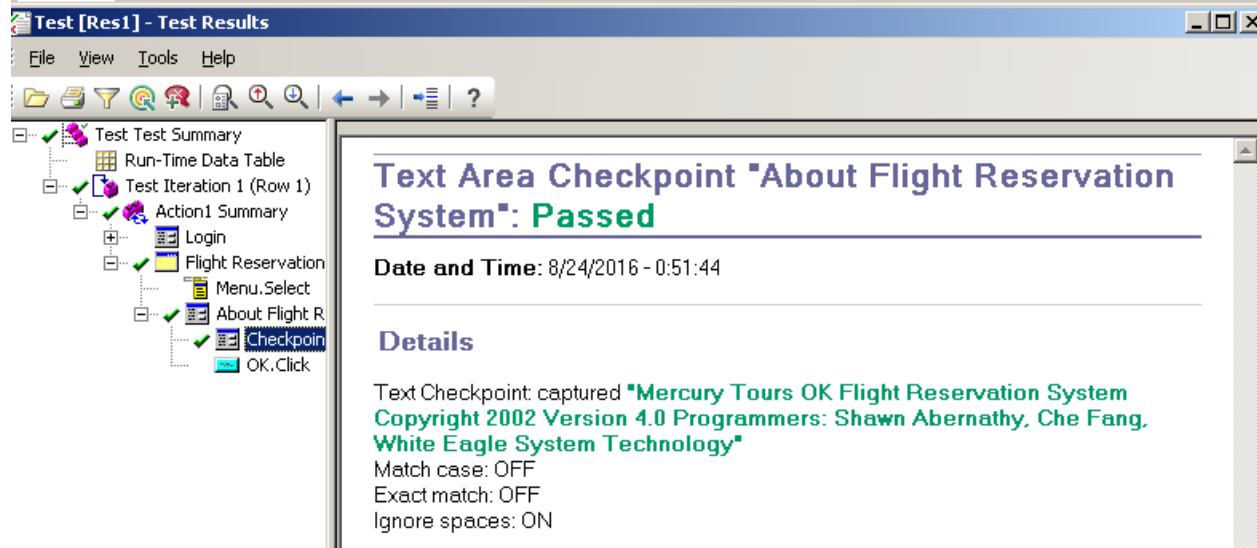
Now click the ok button above screen you will see a checkpoint inserted in your code and also click ok button on flight application screen to complete the login operation. You can also see the existing checkpoint on the design time  
Insert(Design) > Checkpoint > existing checkpoint

**Page Checkpoints** – A standard checkpoint created for a web page becomes a page checkpoint. It is used to check no of links and images on a web page. Page checkpoints can be used to check load time of a page.

1. Open a web page [www.google.com](http://www.google.com)
2. Click on the record button first, this time no recording should be there.
3. Insert > Standard Checkpoint >
4. Click on the page
5. Object selection properties dialog box opens
6. Click on the page object
7. Click Ok Button
8. Page checkpoint properties dialog box opens, keep all default properties selected.
9. The checkpoint is inserted into the script
10. Now stop recording and run the script
11. Click on page checkpoint in results viewer.
12. You can see load time no of images
13. No of links (including broken links)
14. All the links
15. Image names and sources can be seen.

**Text Area Checkpoint** – This is used for windows applications. This compares text string within a defined area according to criteria specified. It is almost similar to text checkpoint with only difference is that text checkpoint work on a control whereas text area checkpoint works on a selected region.

1. Click on the record button first, this time no recording should be there.
2. Launch flight application and enter username and password to login
3. Go to about flight reservation system
4. Click Insert > Checkpoint > select text area checkpoint
5. Drag mouse & select the bottom part where text is written.
6. Object properties dialog selection box opens.
7. Configure text selection properties. Keep them to default here.
8. Stop recording.
9. Run the script.
10. On results viewer go to the checkpoints double click to see the details.
11. It will show the selected text in green if passed.



### A Text Area checkpoint sample

**Bitmap Checkpoint** – Bitmap checkpoint does pixel by pixel comparison of images.

**Difference between image and bitmap checkpoint** – Image checkpoint works on a web environment only whereas bitmap works in any given environment. Bitmap checkpoint can be used to compare an area of an application or page, an object or any part of an object. On setting this checkpoint it captures the chosen portion of the screen as a bitmap and compares it with the result at run time. In contrast, image checkpoint is just for webimage objects.

Therefore, Bitmap checkpoint captures the visible parts of your AUT and compares them as bitmaps, pixel by pixel.

Typically this is used to check maps, logos or any other diagrams in your AUT.

Bitmap checkpoints are dependent on specific values like RGB values, screen resolutions, OS so any changes to these values might affect the checkpoint

QTP cannot capture any part of object that is scrolled of screen or hidden by any other object.

If any app is overlapping your AUT it will be captured in your checkpoint.

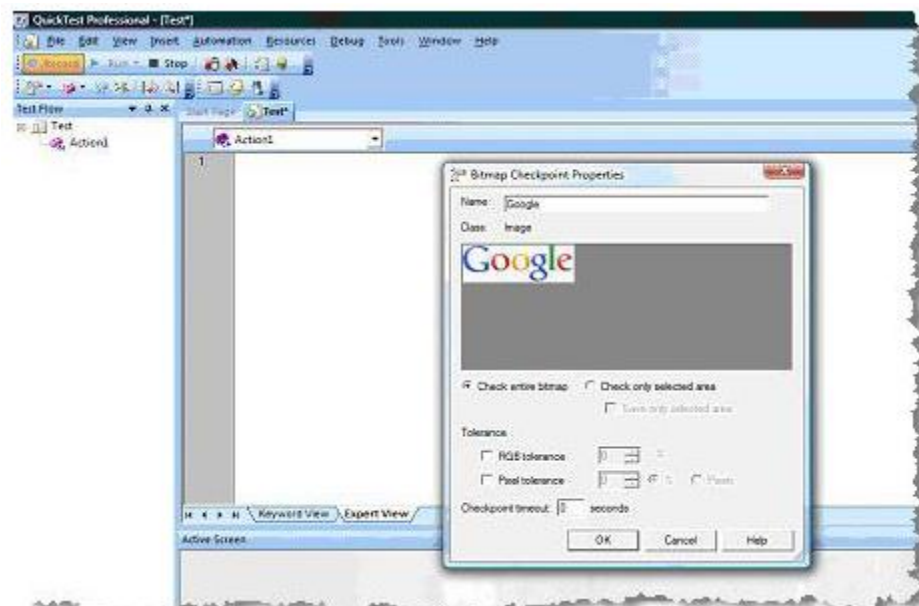
### Steps to create bitmap checkpoint

1. Start recording
2. Open google.com page
3. Insert(Design) > Checkpoint > bitmap checkpoint
4. Then select the google logo.
5. The image appears in bitmap checkpoint properties dialog box
6. Select check full bitmap or check a selected area.

7. If you check the selected area option you will be able to select a given area of image.
8. Keep rest all properties as default.
9. Click Ok
10. The following code will be inserted in expert view.

```
Browser("Gmail: Email from Google").Page("Gmail: Email from Google").Image("Google").Check CheckPoint("Google")
```

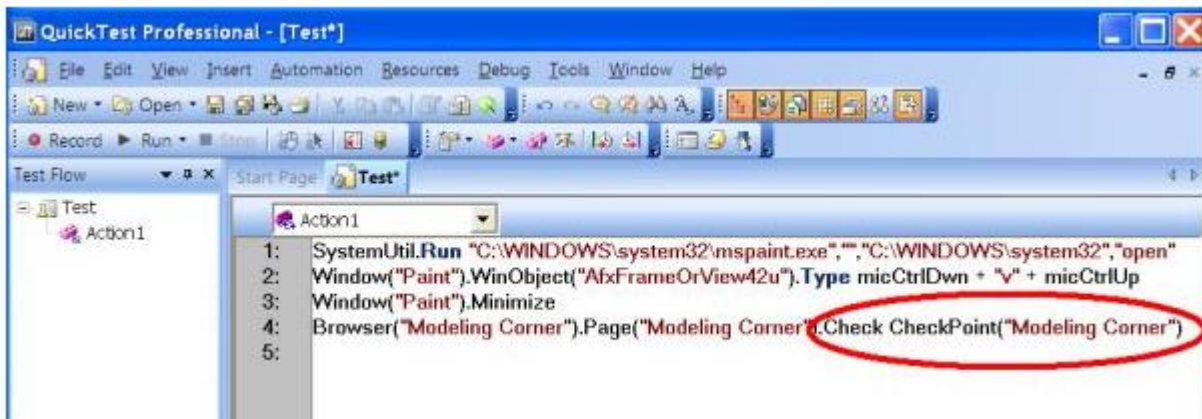
Now run the above script and check the results. If the images changes the new as well as old image will be seen in results viewer.



**Text CheckPoint** – There will be many instances when we need to check the text displayed in the application, one way is that we can use standard checkpoint to check the text property of control.

#### **Steps to create Text checkpoint -**

1. Start recording
2. Open google.com page
3. Insert(Design) > Checkpoint > Text checkpoint
4. Select the text create an account.
5. Text Checkpoint dialog box appears.
6. Keep the default properties.
7. Following line of code is added to script



**Database Checkpoint** – This checkpoint is used to check values in a database table. Let's take an example from flight application and see how this works. We will check here if the name of passenger is updated in orders table when it is changed in application. Steps to insert database checkpoint

- Login to flight application using username and password.
- Select a date and to and from city, select a flight from flight button
- Enter a passenger name e.g Paul.
- Insert order.
- Go to insert > database checkpoint.
- Database query wizard appears
- Select option specify sql query manually.
- Click Next
- Select create on database query wizard
- Select machine data source > select QT32 as data Source
- Enter sql statement as select customer\_name from orders
- Select finish. It will show results of query
- Now go to flight application. Change name to e.g Bill click on update order.
- In your script you can see the checkpoint inserted.
- Go to checkpoint properties (right click in script).
- Change constant value to billy
- Now run the script.

**XML Checkpoint** - You can perform checkpoints on XML documents contained in Web pages or frames, on XML files, and on test objects that support XML.

**We need to install web service addins in QTP.**

An XML checkpoint is a verification point that compares a current value for a specified XML element, attribute and/or value with its expected value.

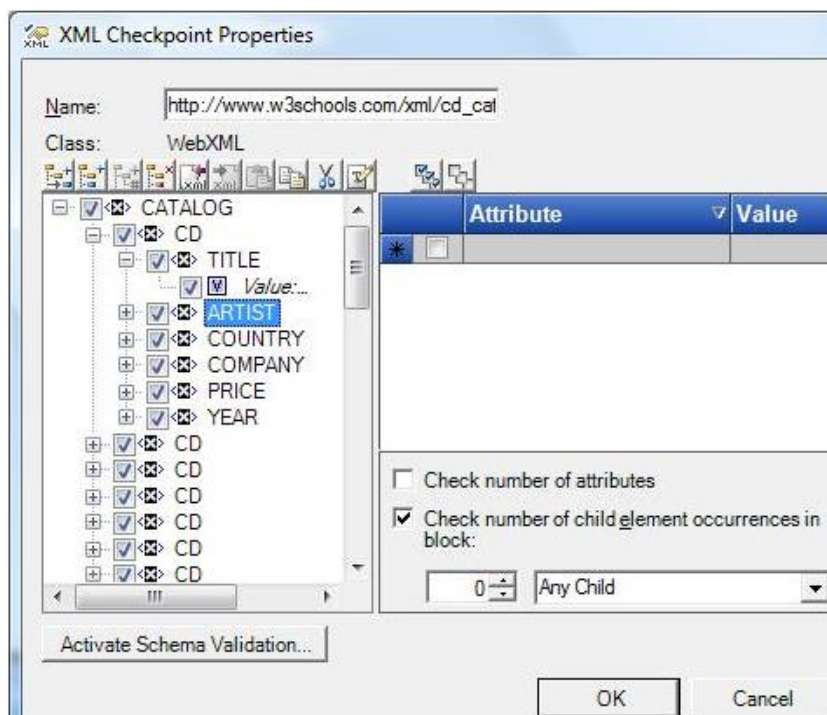
When you insert a checkpoint, QuickTest adds a checkpoint step in the Keyword View and adds a Check CheckPoint statement in the Expert View.

### You can create three types of XML checkpoints:

1. XML Web Page/Frame Checkpoint. Checks an XML document within a Web page or frame.
2. XML File Checkpoint. Checks a specified XML file.
3. XML Test Object Checkpoint. Checks the XML data for an object or operation.

### Steps to create xml checkpoints:

1. Insert > Checkpoint > Xml Checkpoint
2. Select the page where you want to insert the checkpoint
3. The following screen comes up:



4. Set the values as desired in this screen and click OK
5. `Browser("Browser").Page("Page").WebXML("http://www.w3schools.com/xml/c").Check CheckPoint("http://www.w3schools.com/xml/cd_catalog.xml")`
6. Run the script now.

**DataTables** – These are used to parameterize the tests and we need to test with multiple sets of data.

DataTable is just like Microsoft Excel file. The DataTable contains a global sheet and a separate sheet for each action in the test. So we two types of sheets in data table.

1. **Global sheet**
2. **Local sheet**

Global sheet is available to all actions in the test and local sheet is available to the corresponding action. The best practice is to use local data sheet while parameterizing the test.

**Design Time DataTable:** While creating editing the test scripts, you enter data directly in the data table displayed there. You can view data table by selecting *View >> Data Table* option. This data table is called *Design Time DataTable*.

**Runtime DataTable:** The data table used while running the test is called Runtime DataTable. The data in the runtime data table may be same as design time data table and/or can be entered by importing data from excel sheet, text file etc. The runtime data table can be seen in the Test Results window when run session is ended.

#### **DataTable Parameter:**

Each column in global/local sheet of datatable is called a Parameter. We can rename the parameter just by double clicking on the column header and giving the name to it. Data can be entered in the column/parameter simply by clicking on the cell and entering the value.

### **How to get Child Objects In QTP?**

The objects (text box, combo box, links) contained in the frame or Window is known as child objects.

The below Script gets the all the name of the links from the website "[www.easycalculation.com](http://www.easycalculation.com)"

```
Dim oDesc
Set oDesc = Description.Create
oDesc("micclass").value = "Link"

'Find all the Links
Set obj = Browser("Math Calculator").Page("Math Calculator").ChildObjects(oDesc)

Dim i
'obj.Count value has the number of links in the page
For i = 0 to obj.Count - 1
    'get the name of all the links in the page
    x = obj(i).GetROProperty("innerHTML")
    print x
Next
```

