**UFT Class 13** (Parameterization)   
----------------  
**What is Parameterization?**  
  
Replacing constant (Fixed) values with parameter is called Parameterization.  
-----------------------------------------------  
**Purpose of Parameterization**  
  
    Passing single value but different value every time  
  
    Passing multiple values.  
----------------------------------------------  
**What is Data Driven Testing?**  
  
Testing the same functionality using multiple sets of Test Data.  
---------------------------------------  
**Why Data Driven Testing?**  
  
    For Negative Testing  
    For checking the reliability of the System.  
----------------------------------------  
**Relation between Parameterization and Data driven Testing**  
  
    We use Parameterization feature in Data driven Testing.  
  
Data Driven Testing means passing multiple values only.  
  
Parameterization means not only passing multiple values also passing single value.  
-----------------------------------------------------------  
**How to Parameterize?**  
  
We can use either Variables or Constants or Arguments or Environment variables  
or Data Table parameters or Action parameters or Dictionary keys as Parameters.  
  
Generally, If it is Test Level we prefer variables, Function level we prefer Arguments.  
-----------------------------------------------  
**Ways of Parameterization**  
------------------------------------  
i) Generate and Pass values using Loop statements  
  
ii) Dynamic submission of Test Data  
  
iii) Using Data Table Parameters  
    a) Enter test data directly into Data Table and connect to the Test.  
    b) Import test data from external files (text/excel) and connect to the test.  
    c) Import test data from a database and connect to the test.  
    d) Use Data Table methods and programmatic statements.  
  
iv) Using Environment variables  
  
v) Using Action Parameters  
  
vi) Using Function Arguments  
  
vii) Using Dictionary object (Keys)  
  
viii) Using Automation objects  
    a) Read data directly from a text file and connect data (Filesystem object)  
    b) Read data directly from an excel file and connect data (Excel Application object)  
    c) Read data directly from a database and connect data (Database Objects)  
-------------------------------------------------  
**i) Generate and Pass values using Loop statements**  
  
a) We can generate Sequential numbers (1, 2, 3......)  
b) We can generate Logical numbers (2, 4, 6....10, 20, 30....)  
  
**Limitations:**  
  
we can't desired numbers (1, 7, 34, 56.....)  
  
We can't generate string type data (User name, Password, city name etc....)  
------------------------------------------  
'Open 1 to 5 records  
For OrderNumber = 1 To 5 Step 1  
Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNumber   
wait 2  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Next  
---------------------------------------  
'Open even numbers up to 10  
For OrderNumber = 2 To 10 Step 2  
Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNumber

wait 2  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Next  
-------------------------------------  
'Open first 5 orders in reverse order  
For OrderNumber = 5 To 1 Step -1  
Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNumber

wait 2  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Next  
------------------------------------  
'Open first 5 orders except 4th order  
For OrderNumber = 1 To 5 Step 1  
If OrderNumber <> 4 Then  
Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON" Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNumber wait 2  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").

End If  
Next  
-------------------------------------------------  
**ii) Dynamic Submission of Test Data**  
  
> No resource (Test data file) is required, but not recommendable for large amount of test data.  
  
> User interaction is mandatory for every test iteration.  
  
Ex:  
For i = 1 To 3 Step 1  
Agent = InputBox("Enter Agent Name")  
Password = InputBox ("Enter Password")  
SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe","","C:\Program Files\HP\Unified Functional Testing\samples\flight\app\",""  
Dialog("Login").Activate   
Dialog("Login").WinEdit("Agent Name:").Set Agent Dialog("Login").WinEdit("Password:").Set Password

Wait 2  
Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").Close  
Next  
---------------------------------  
It is used to submit captcha fields.  
----------------------------------  
**iii) Using Data Table Parameters**  
  
> It is an integrated spread sheet for Data related operations.  
  
Navigation:  
  
View menu -> Data  
  
> Data table has 2 types of sheets   
  
a) Global sheet   
  
b) Action sheets (Local sheets)  
  
> Data Table located in 2 areas:  
  
a) In UFT Tool main window (Design time Data table)  
  
b) In UFT Tool Result window (Run-time Data Table)  
  
> Usage of Data Table  
------------------------------------------------  
Steps for Data Driven Testing using Data table  
  
a) Generate the basic Test and run it once.  
  
b) Launch Data table and enter or Import Test data.  
  
c) Connect test data to the Test.  
  
    Using editor view Or Using Keyword view or Using Data Driver  
  
d) Run the Test  
-------------------------------------------

**UFT Class 14**

(Parameterization Part 2 - Parameterization using Data Table Parameters)

**3) Using Data Table Parameters**

**What is Data Table?**

It is an integrated spread sheet for Data related operations

**How to launch:**

View -> Data

> Data Table has 2 types of sheets

a) Global sheet

b) Actions sheets (Local sheets)

> Data Table located in 2 areas

a) In UFT tool main window (Design-time Data Table)

b) UFT tool Result window (Run-time Data Table)

> It attaches one Data Table for every new Test

**Usage of Data Table:**

a) Enter Test data directly into Data Table and connect to the Test.

b) Import data from external files (text or excel) and into Data Table and connect to the Test.

c) Import data from a Database and into Data Table and connect to the Test.

d) Using Data table methods and programmatic statements

**Steps for Data Driven Testing using Data Table:**

a) Generate the basic Test and run it once

b) Launch Data table and Enter or Import test data.

c) Connect test data to the Test

                Using Editor View

                Or

                Using Keyword view

                or

                Using Data Driver

d) Run the Test

-------------------------------

**Syntax for connecting Test Data to the Test using Editor view**

DataTable("Column Name"/Column Id, Sheet Name / Sheet Id)

Ex:

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set DataTable(1, 1)

Dialog("Login").WinEdit("Password:").Set DataTable(2, 1)

Wait 2

Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").Close

---------------------------------------

**Import Test Data from External files (Text or Excel)**

i) Text file

> Data Table considers Text file first line as Column headers

> In between two columns Tab space is the delimiter

---------------

ii) Excel File

> Data Table considers Excel sheet first row as Column headers

------------------------------------------------------------------

**c) Import data from a Database and into Data Table and connect to the Test.**

Steps:

> Create a Database

> Create Tables

> Enter Data

---------

> Create DSN (Data Source Name)

Note: If you want to connect to a Database then Connection string (DSN) is required

> Connect Test Data

------------------------------

**Creating DSN:**

> Launch Data Table

> Place mouse pointer on Data Table and Right Click

> Sheet > Import > From Database

> Select "Specify SQL statement manually" option

> Click Next

> Click Create

> Click New

> Select Driver for our Database and Click Next

> Browse path to store DSN

> Click Next > Click Finish

> Click Select

> Browse database and select

> Click OK > Click OK

> Enter SQL statement

> Click Finish

-------------------------------------

**d) Using Data table methods and programmatic statements**

Note: We insert programmatic statements in order to overcome some limitations of Tool features.

Ex:

'Create a new sheet in Run-time Data Table

DataTable.AddSheet "Login"

'Import Test data from an external file

Datatable.ImportSheet "C:\Users\G C Reddy\Desktop\abcd.xls", 1, "Login"

RowsCount = DataTable.GetSheet("Login").GetRowCount

For i = 1 To RowsCount Step 1

DataTable.SetCurrentRow(i)

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set DataTable("Agent", "Login")

Dialog("Login").WinEdit("Password:").Set DataTable(2, 3)

wait 2

Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").Close

Next

-------------------------------------------------

**'Data Driven Testing using Data Table parameters and programmatic statements**

DataTable.AddSheet "Login"

DataTable.ImportSheet "C:\Users\G C Reddy\Desktop\abcd.xls", 1, 3

RowCount = DataTable.GetSheet("Login").GetRowCount

For i = 1 To RowCount Step 1

                Datatable.SetCurrentRow(i)

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set DataTable(1, 3)

Dialog("Login").WinEdit("Password:").Set DataTable(2, 3)

wait 2

Dialog("Login").WinButton("OK").Click

If Window("Flight Reservation").Exist(12) Then

Window("Flight Reservation").Close

DataTable(3, 3) = "Login Successful -Passed"

Else

SystemUtil.CloseDescendentProcesses

DataTable(3, 3) = "Login Unsuccessful -Failed"

End If

DataTable.ExportSheet "C:\Users\G C Reddy\Desktop\Result1.xls", "Login"

Next

------------------------------------

**'Read Data from external file and write Result and error messages**

Dim RowCount, OrderNumber

'Data Driven Testing using Data Table parameters and programmatic statements

DataTable.AddSheet "Login"

DataTable.ImportSheet "C:\Users\G C Reddy\Desktop\abcd.xls", 1, 3

RowCount = DataTable.GetSheet("Login").GetRowCount

For OrderNumber = 1 To RowCount Step 1

                Datatable.SetCurrentRow(OrderNumber)

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set DataTable(1, 3)

Dialog("Login").WinEdit("Password:").Set DataTable(2, 3)

wait 2

Dialog("Login").WinButton("OK").Click

If Window("Flight Reservation").Exist(12) Then

Window("Flight Reservation").Close

DataTable(3, 3) = "Login Successful -Passed"

Else

DataTable(4, 3) = Dialog("Login").Dialog("Flight Reservations").Static("Agent name must be at").GetROProperty("text")

SystemUtil.CloseDescendentProcesses

DataTable(3, 3) = "Login Unsuccessful -Failed"

End If

DataTable.ExportSheet "C:\Users\G C Reddy\Desktop\Result1.xls", "Login"

Next

**UFT Class 15**(Actions in UFT)  
--------------------------  
**Actions**  
  
**What is Action?**  
  
Test- One or more actions to perform operations  
  
Action - Set of statements or steps to perform operations  
  
Statement/step - a minimal executable unit  
  
Note: One test may have one or more actions  
  
One statement / step may have one or more lines and vice versa  
  
**Purpose of Actions**  
  
                  Test1 (1 Action)    Test2 (3 Actions)  
------------------------------------------------------  
Performance    High               Low  
-----------------------------------------------------  
\* Reusability    Less/No          More  
---------------------------------------------------  
Readability      Low                High  
------------------------------------------------  
Locating Errors  Diff              Easy  
-------------------------------------------------  
**Types of Actions**  
**a) Reusable Action**  
It can be used multiple times in the Home tests and from other Tests  
  
It can be called a s well as copied  
  
Calling - Only execution, we can't edit the code  
(Centralized maintenance of Actions)  
  
Coping - Execute and Edit the code  
(No Centralized maintenance of Actions)  
  
**b) Non-reusable Action**  
It can be used only once in the Home Test  
  
It can't be called but copied  
---------------------  
**c) External Action**  
It is nothing but Reusable in the Home Test, calling Test point of view External.  
  
    Test1                                         Test2  
--------------------------------------------------------  
  Login(Reusable)    ------------>    Login (External Action)          
  Insert Order(Non-reusable)         Open Order(Reusable)  
                                                    Fax Order(Non-reusable)  
------------------------------------------------------  
Operations on Actions  
    Create New Action  
    Rename Action  
    Delete Action  
    Call to Existing Action  
    Copy Action  
    Make Reusable Action as Non-reusable and vice versa  
    Action Parameters  
        Input parameters  
        Output parameters  
  
**a) Create New Action**  
**Navigation:**  
  
Create New Test  
  
> Design menu > Call to New Action  
  
> Enter Action Name (If you don't want the default Name)  
  
> Write Description (Optional)  
  
> If want to create Non-reusable Action then Uncheck Reusable Action check box  
  
> Click OK  
-------------------------------  
**Creating a Test using multiple Actions**  
Create New Test  
  
> Create New Actions (One by one)  
  
> Generate script for every Action  
---------------------------------------  
**b) Rename Action**  
Select the Test  
  
> Select required Action and Right click  
  
> Select "Action Call properties"  
  
> Change the Name  
  
> Click OK  
  
> Conform  
--------------------------------------  
**c) Make Reusable Action as Non-reusable and vice versa**  
Note: We can create Reusable or Non-reusable actions while creating the actions or  
After creating the Action.  
      
**Reusable to Non-Reusable:**  
Select the Test in UFT Editor View  
  
> Select required Action and Right click  
  
> Select Action Properties  
  
> Uncheck "Reusable Action" Check box  
  
> Click Ok  
  
> Conform  
-----------------------------------  
**Non-Reusable to Reusable**  
Select the Test in UFT Editor View  
  
> Select required Action and Right click  
  
> Select Action Properties  
  
> Check "Reusable Action" Check box  
  
> Click Ok  
  
> Conform  
--------------------------  
**d) Call to Existing Action**  
Design menu  
  
> Call to Existing Action  
  
> Browse path of the Test  
  
> Select required Action  
  
> Ok  
----------------------------------  
**e) Call to copy of Action**  
Design menu  
  
> Call to copy of Action  
  
> Browse path of the Test  
  
> Select require Action  
  
> OK  
----------------------------------  
**f) Delete Action**  
Select the Test  
  
> select required Action and Right click  
  
> Select "Delete Action"  
  
> OK  
------------------------------------------  
**g) Action Parameters**  
**Input parameters**  
To pass input data, we can pass single value or multiple values  
  
Purpose of Action Input parameters:  
  
If we don't use input parameters then fixed input data for every calling Action  
If we use input parameters then we can pass different values and multiple values also.  
  
**How to Create Action input parameters**  
Select the Test  
  
> Select Action and Right click  
  
> Select Action Properties  
  
> Select "Parameters" tab  
  
> Add Input parameters one by one and set default value (If required)  
  
> Ok  
---------------------  
**How to connect:**  
In Editor view  
  
Type Parameter("ParameterName")  
-----------------------------------------------  
Assignment:  
  
Give an example for Action Output Parameters  
  
To return the Test result.

**UFT Class 16**(Environment Variables)  
-----------------------  
**What are Environment Variables?**  
They are global variables depends on operating system environment.  
-------------------------------------  
**Purpose of Environment Variables**  
    Centralized Maintenance of Variables  
  
    Tests portability  
--------------------------------------------------  
Note 1: Whenever we want to use any variable in multiple Tests then choose Environment variables.  
  
Note 2: Whenever you want use any variable in a single Test then no need to choose Environment variables.  
  
Tests portability:  
  
Executing Tests from different machines.  
--------------------------------------  
**Types of Environment Variables**  
a) Built in variables  
  
b) User defined variables  
  
-----------------------------  
**Steps:**  
i) Create Environment (User defined) variables and export to XML  
  
ii) Associate Environment Variables file  
Or  
Load Environment Variables file during execution  
  
iii) Access environment Variables  
Syntax:  
Environment("EnvironmentVariablename")  
  
Ex:  
Msgbox Environment("OS")  
Msgbox Environment("ProductDir")  
  
  
iv) Modify Environment variables (If required)  
  
v) Delete Environment Variables (Optional)  
---------------------------------------  
Note: Environment variables are case sensitive.  
-----------------------------------------------  
**1) Navigation for Creating Environment Variables**  
File menu  
> Settings  
> Environment  
> Select variable type as "User defined"  
> Click Add icon  
> Enter Variable name and value  
> Click OK   
(Again click Add icon if we want to create multiple variables)  
  
> Click Export   
  
> Browse path to store XML file  
  
> OK  
-----------------------------  
**2) Navigation for Associating Environment Variables file**  
File menu  
  
> Settings  
  
> Environment  
  
> select Variable as "User defined"  
  
> Check "Load variables and values from external file"  
  
> Browse path of the XML file  
  
> Apply > OK  
-------------------------------  
**Example:**  
SystemUtil.Run Environment("AppPath")  
Dialog("Login").Activate   
Dialog("Login").WinEdit("Agent Name:").Set Environment("Agent")  
Dialog("Login").WinEdit("Password:").Set Environment("Password")   
Wait 2  
Dialog("Login").WinButton("OK").Click   
--------------------------------------------------------  
**'Load Environment Variables file during execution**

Environment.LoadFromFile "C:\Users\G C Reddy\Desktop\abcd.xml"  
---------------------------------------------------------  
Note: without UFT tool also we can create Environment variables  
  
> Launch Notepad  
  
> Type tags  
  
> Sava as XML file  
--------------------------  
**Tags of Environment variables file**  
Environment open tag end tag  
  
Variable open and close tag  
  
Name open and close tag  
  
Value open and close tag  
------------------------------  
**Example:**  
<Environment>  
    <Variable>  
        <Name>Agent</Name>  
        <Value>Chennai</Value>  
    </Variable>  
      
</Environment>  
-------------------

**UFT Class 17**

(Synchronization, Checkpoints Part-1)

**Synchronization UFT with AUT**

**What is Synchronization?**

UFT        AUT

It is a process of matching the speeds of UFT tool and AUT, in order to get proper execution and Result.

**Why Synchronization?**

During Test execution UFT gives instructions one by one with same speed, but AUT takes less time for some steps execution

and more time for some steps execution, in order to keep them in sync then Synchronization is required.

**When Synchronization is required?**

UFT tool default Synchronization time is 20 seconds, if any step takes more than 20 seconds for execution then Synchronization is required.

**Types of Synchronization**

a) Conditional Synchronization

b) Unconditional Synchronization

**How to Synchronize UFT and AUT?**

**a) Insert wait statement**

Syntax:

Wait (Time in seconds) or Wait Time in seconds

Ex:

Wait (10)

Note: It is conditional Synchronization, waits maximum time eventhough operation is completed.

**b) Insert Synchronization point**

To insert Synchronization point, identify the Object and property on which we need to insert Synchronization point.

Ex:

"Delete Order" Button is Object

Enabled property

Syntax:

Object.WaitProperty "PropertyName", Value, Time in milli seconds (Time is optional)

Note: It waits required time only.

Example:

Window("Flight Reservation").WinButton("Delete Order").WaitProperty "enabled", True, 30000

Navigation:

Place cursor in desired location

> Select Record mode

> Design menu

> Synchronization point

> Show the object

> Select Property

> Enter Property value

> Set Time in mille seconds > OK

----------------------------------------

**c) Increase Tool default time**

Navigation:

File menu

> Settings

> Run

> Select Object synchronization Timeout

> Chang the value

> OK

-------------------------------------

Note: If we increase tool default time, that settings applicable for all steps in that test.

-----------

**d) Using Exist property**

'Exist property for Synchronization

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe","","C:\Program Files\HP\Unified Functional Testing\samples\flight\app\",""

Dialog("Login").Activate  
Dialog("Login").WinEdit("Agent Name:").Set "asdf"

Dialog("Login").WinEdit("Password:").SetSecure "54f7bb545b92f523dfa596ccfead0af7e4459d54"

Dialog("Login").WinButton("OK").Click

If Window("Flight Reservation").Exist(10) Then

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"  
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "1"

Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

End If

--------------------------------

'Exist property for Error Handling

If Not Dialog("Login").Exist(3) Then

SystemUtil.Run "C:\Program Files\HP\Unified Functional Testing\samples\flight\app\flight4a.exe","","C:\Program Files\HP\Unified Functional Testing\samples\flight\app\",""

End If

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set "asdf"

Dialog("Login").WinEdit("Password:").SetSecure "54f7bb545b92f523dfa596ccfead0af7e4459d54

Dialog("Login").WinButton("OK").Click

----------------------------------------

**Select an appropriate method**

----------------------------------

**Advantages of Wait statement:**

a) It won't corrupt

b) No object reference is required

c) In order to observe the Test execution process then wait statement is the only solution

d) In order to capture screen shots during execution then wait statement is useful.

e) It can be used internally as well as externally

f) It is the only solution for Non-recordable steps.

-------------------------------------------------------

Note: UFT Tool default synchronization Time is only for Recordable Steps

It can't be used for Non-recordable steps

Ex:

Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "1"

Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Window("Flight Reservation").WinEdit("Tickets:").Set "5"

Window("Flight Reservation").WinButton("Update Order").Click

Wait 9

Message = Window("Flight Reservation").ActiveX("Threed Panel Control").GetROProperty("text")

If Message = "Update Done..." Then

                Result = "Order Updated - Passed"

                Else

                Result = "Order Not Updated - Failed"

End If

Msgbox Result

--------------------------------------------------------------------

**Insert Checkpoints**

---------------------

**What is Checkpoint?**

It is a verification point, takes expected result from the User and compares with actual results during test execution and provides Test Result.

Expected Result: User (Tester)

Actual Result: During Test Execution

Test result: UFT provide test result after comparing expected with actual.

**Types of Checkpoints**

                1) Standard Checkpoint

                2) Text Checkpoint

                3) Text area Checkpoint

                4) Bitmap Checkpoint

                5) Database Checkpoint

                6) Accessibility Checkpoint

                7) XML Checkpoint (From Resource)

                8) XML Checkpoint (from Application)

                9) File Content Checkpoint

                10) Page Checkpoint

                11) Image Checkpoint

                12) Table Checkpoint

**Drawbacks of Checkpoints**

-------------------------------

**UFT Class 18** (Inserting Checkpoints Part -1)  
-----------------------------  
UFT Test Process  
    Enhancing Tests  
        Inserting Checkpoints  
  
**What is Checkpoint?**  
It is a verification point, takes expected result from the User(Tester), compares with actual Results and provides Test Result.  
  
Types of Checkpoints (9 + 3)  
  
1) Standard Checkpoint  
  
It checks object properties values.  
-------------------------------  
2) Text Checkpoint  
  
It checks text value of an object in different ways  
------------------------------------  
3) Text area Checkpoint  
  
It checks text area present in the Application in different ways  
-----------------------------------  
4) Bitmap Checkpoint  
  
It compares two bitmaps  
-----------------------------------  
5) Database Checkpoint  
  
It checks content of a Database Table cell wise.  
-------------------------------------  
6) Accessibility Checkpoint  
  
It checks weather the web page was developed according to W3C standards or not?  
---------------------------------  
7) XML Checkpoint (From Application)  
  
It checks content of XML page.  
----------------------------------  
8) XML Checkpoint (From Resource)  
  
It checks content of XML file.  
-----------------------------------  
9) File content Checkpoint  
  
It checks content of a text file.  
------------------------------------  
10) Page Checkpoint  
  
It checks number of Links, number of images in a webpage and creation time of the Page.  
-----------------------------------  
11) Image Checkpoint  
  
It checks Image object Properties and values.  
----------------------------------  
12) Table Checkpoint  
  
It checks content of a web table cell wise.  
----------------------------------------------  
Note 1: from 6 to 12 checkpoints (except File content and XML (From Resource) checkpoints) are only for Web applications  
  
Note 2: From 10 to 12 Checkpoints are hidden Checkpoints, using Standard Checkpoint we can insert them.  
  
Note 3: Database, File Content and XML (From Resource) Checkpoints are Back-end Checkpoints  
  
Note 4: We can insert multiple Checkpoints in a Test.  
  
Note 5: We can Edit and delete Checkpoints.  
-------------------------------------------------------  
We can insert standard checkpoints using Editor view or Using Keyword view or using Active screen.  
  
Insert Standard Checkpoint:  
  
Identify the Object and Property on which we need insert Standard checkpoint.  
  
Select Record mode  
  
> Design menu  
  
> Checkpoint  
  
> Standard Checkpoint  
  
> Take hand icon and show the object  
  
> Select Property  
  
> Enter expected result > OK  
---------------------------------  
Edit Standard Checkpoint:  
  
Select Checkpoint statement and Right click  
  
> Modify the value  
  
> OK  
------------------------------  
Delete Standard Checkpoint:  
  
> Select Checkpoint statement and right click  
  
> Delete  
--------------------------------------------------  
Ex:  
VbWindow("CALCULATOR").Activate   
VbWindow("CALCULATOR").VbEdit("VALUE1").Set "10"   
VbWindow("CALCULATOR").VbEdit("VALUE2").Set "20"   
  
VbWindow("CALCULATOR").VbButton("ADD").Click   
VbWindow("CALCULATOR").VbEdit("RESULT").Check CheckPoint("RESULT")  
  
VbWindow("CALCULATOR").VbButton("SUB").Click   
VbWindow("CALCULATOR").VbEdit("RESULT").Check CheckPoint("RESULT\_2")   
  
VbWindow("CALCULATOR").VbButton("MUL").Click   
VbWindow("CALCULATOR").VbEdit("RESULT").Check CheckPoint("RESULT\_3")   
  
VbWindow("CALCULATOR").VbButton("DIV").Click   
VbWindow("CALCULATOR").VbEdit("RESULT").Check CheckPoint("RESULT\_4")  
  
VbWindow("CALCULATOR").VbButton("CLEAR").Click   
--------------------------------------------  
 Configure Accessibility Checkpoint  
  
Tools menu  
  
> Options  
  
> GUI Testing  
  
> Web > Advanced  
  
> Check / Uncheck check boxes  
  
> OK  
---------------------------------  
Result Criteria:  
  
Check item available and according W3C Standards then pass  
  
Check item available and Not according W3C Standards then fail  
  
Check item Not available then pass