**Object Identification:**

Mandatory properties

Assistive properties

Visual Relation Identifier: Hortizontal,Vertical, Distance and Hierarchy(Referencing to nearest objects)

Smart Identification: Base Filter Properties,Optional Filter Properties

Ordinal Identifier: Index(0,1..), Location(1,2,..), Creation Time(0,1,..) for Browser

Virtual Objects: Mapping to Standard objects

**Recording Modes:**

Normal:Default

Low-level: records the co-ordinates in the application, Recording done in the format of Window,WindowObj (Used in case of Obj not identified)

Analog :exact mouse and key strokes that the user performs in relation to either the screen or AUT (Ex: In case of insert signatures)

Window/app.RunAnalog “Track1”

InsightObject Recording

**Actions:**

Reusable – only these can be called multiple times within the same or different tests

Non-reusable – These cannot be called by any other actions

External actions – a reusable action when called from another action becomes an external action in the called action.

RunAction “Action Name”,Oneiteration(0)/ alliterations(1)/”1-4”, i/p parameters, o/p parameters

ExitAction(1)

LoadAndRunAction “TestPath” ,“Action Name”,Oneiteration(0)/ alliterations(1)/”1-4”, i/p parameters, o/p parameters

**Checkpoints:**

Standard Checkpoint

Image Checkpoint-Web Image

Bitmap Checkpoint

Page Checkpoint- Checking links,Sources of images,Check for broken links

Text Checkpoint

XML Checkpoint

Text Area Checkpoint

Table Checkpoint-Web table

Accessibility Checkpoint- Active X,Alt property,Applet,Frame titles,Multimedia,Server side Image,Tables

Database Checkpoint

**Mercury Timer:**

Continue Method  - Continue counting  
Reset Method - Reset timer  
Start Method - Starts Timer  
Stop Method - Stops Timer

ElapsedTime Property

MercuryTimers("Timer1").Start

**Insert Transactions:**

Used to calculate time difference between transactions

Services.StartTransaction "Name"

Services.EndTransaction "Name"

**Data Table:**

datatable.SetCurrentRow(i)

datatable.GetRowCount

DataTable("ColumnName", dtLocalSheet/dtGlobalSheet)

DataTable.AddSheet(SheetName)

DataTable.ExportSheet(FileName,SheetName)

DataTable.GetSheet(SheetID)

DataTable. ImportSheet(FileName, SheetSource, SheetDest)

**Runtime Loading:**

Function Library:

LoadFunctionLibrary "FullFilePath"

ExecuteFile "C:\Functions\SampleFunction.vbs"

objQTP.Test.Settings.Resources.Libraries.add “Path”

OR:

RepositoriesCollection.Add(SORpath) ‘It will add to Action1

qtApp.Test.Actions("ActionName").ObjectRepositories.Add "C:\Program Files\Mercury Interactive\QuickTest\Repository2.tsr"

Recovery:

Set qtTestRecovery = qtApp.Test.Settings.Recovery.Add"E:\Recoveryfiles\sample.qrs", " loginPop ", 1

**Descriptive Programming:**

Set obj\_Desc = Description.Create

obj\_Desc(“name”).value= “txt.\*”  
obj\_Desc(“name”).regularexpression= “txt.\*”

Set allCheckboxes = Browse(“Browser”).Page(“Page”).ChildObjects(obj\_ChkDesc)

TestObject("PropertyName1:=PropertyValue1", "..." , "PropertyNameX:=PropertyValueX")

**Xpath and CSS:**

Method1: Browse(“Browser”).Page(“Page”).Button("xpath/CSS:=//input[@id=’button’]")

Method2: Directly adding the value to xpath property in object repository

**Object Properties:**

SetTOProperty: Sets the specified value of an identification property for an object

GetROProperty: retrieves the current property value of the object in the application during the test run

GetTOProperty retrieves the values of only those properties that are included in the test object description in Object Repository by QTP.

set TOProperties =Window("Flight Reservation").WinRadioButton("Business").GetTOProperties

sName=TOProperties(i).Name ' Name of the property

sValue=TOProperties(i).Value 'Value of the property

isRegularExpression=TOProperties(i).RegularExpression

**Error Handling:**

On Error Resume Next:Skip the Error

On error Goto 0: turn off the error handling

Err.Number

Err.Description

Err.Clear

**Synchronization:**

WaitProperty

Exist

Wait

Sync

Default Object sync time out-20 Seconds

Default Browser navigation Time out-60 Seconds

**File Format:**

.vbs,.txt,.qfl-file extesion for library file

.mts-script file

.mtr-peraction repository

.tsr-Shared repository

.mtb-batch test

.mst-Action file

.txt,.ini,.xml-Environment Variable File

.qsr- Recovery Scenario

**Shot cut Keys:**

Run : F5  
Run now : Shift + F5  
Stop : F4  
Run from step : Ctrl + F5  
Step Into : F11  
Step Over : F10  
Step Out : Shift + F10

Record : F3  
Stop Run/Recording : F4  
Analog Recording : Ctrl + F3  
Low Level Recording : Shift + F3  
Insert Breakpoint : F9  
Remove Breakpoint : Ctrl + F9  
Clear All Breakpoints : Shift + Ctrl + F9

Open Object Repository : Ctrl + R

ALM Connection : Ctrl + Q

Comment : Ctrl + M  
Uncomment : Ctrl + Shift + M

Step Generator : F7  
Check Syntax : Ctrl + F7

Standard Checkpoint : F12

New Test : Ctrl + N  
New Business Component : Ctrl + Shift + N  
Function Library : Alt + Shift + N  
Application area : Ctrl + Alt + N

Open Test : Ctrl + O  
Open Business Component : Ctrl + Shift + O  
Open Function Library : Alt + Shift + O  
Open Application area : Ctrl + Alt + O

**Optional Step:**

OptionalStep.Browser( ).Page( ).Dialog("Warning").WinButton("OK").Click

[**Ways to Launch your application**](http://mercuryquicktestprofessional.blogspot.com/2006/11/ways-to-launch-your-application.html)**:**

SystemUtil.Run ( FileName, Parameters, Path, Operation )

InvokeApplication "C:\Program Files\Internet Explorer\IEXPLORE.EXE http://www.yahoo.com"  
oShell = CreateObject ("Wscript.shell")

oShell.run "F:\jdk1.3.1\demo\jfc\SwingSet2.bat"'

SystemUtil.CloseProcessByName ‘To close Application

**Using System DLL in UFT:**  
Extern.Declare micHwnd, "FindWindow", "user32.dll", "FindWindowA", micString, micString 'Declare FindWindow method  
hwnd = Extern.FindWindow("Notepad", vbNullString) ‘Get HWND of the Notepad window

**Using .Net DLL in UFT:**

Set oCustom = DOTNetFactory.CreateInstance("oNamespace.ClassName", "C:\MyLib.dll")

**Register User Function:**

RegisterUserFunc "Link", "Click", "MyClick"

UnRegisterUserFunc "Link", "Click"

MyClick (obj, x)

**Dictionary Object:**

Set dict = CreateObject(“Scripting.Dictionary”)

dict.Add “Country”, “India”

If dict.Exists(“Country”) Then

i = dict.Items ‘ Get the items.  
k = dict.Keys ‘ Get the keys.

dict.Remove(“Website”)

dict.RemoveAll ‘ Clear the dictionary

**Create Object:**

Set Var = CreateObject(servername.typename [, location])

CreateObject ("Wscript.shell")

**ADODB:**

Set con=createobject("adodb.connection")

Set rs=createobject("adodb.recordset")

rs.open "SELECT count(\*) FROM [Status$] where Status = 'Failed' ",con

For i=0 to rs.recordcount-1

   SQLExpectedData=rs.fields(Colname)  
   rs.movenext  
Next

con.open "DRIVER={Microsoft Excel Driver (\*.xls)};DBQ=C:\TestStatus.xls;Readonly=True"

con.open "Driver={Microsoft Access Driver (\*.mdb)};Dbq=C:\mydatabase.mdb;Uid=Admin;Pwd=;"

con.open"Driver={SQL Server};server=MySqlServer;uid=MyUserName;pwd=MyPassword;database=pubs"

con.open "Driver={Microsoft ODBC for Oracle};Server=QTPWorld; Uid=your\_username;Pwd=your\_password;"

con.open"Driver={MySQL ODBC 3.51 Driver};Server=localhost;Database=myDB;User=Uname;Password=Pwd;Option=3;"

con.open"Driver={SYBASE SYSTEM 11};Srvr=myServerAddress;Uid=Uname;Pwd=Pwd;Database=myDataBase;"

**Excel:**

set objexcel = Createobject(“Excel.Application”)

objexcel.visible = true

objexcel.DisplayAlerts = False  
Set objWorkbook = objExcel.WorkBooks.Open(“path of the file.xls”)  
Set objDriverSheet = objWorkbook.Worksheets(“name of the sheet / Id of the sheet”)  
columncount = objDriverSheet.usedrange.columns.count  
rowcount = objDriverSheet.usedrange.rows.count

objWorkbook.SaveAs “Path”

**Reading and Writing Data to Notepad:**

**UFT-ALM Integration:**

QCUtil.CurrentRun.ID

GetALMTestPath = QCUtil.CurrentTest.Field("TS\_SUBJECT").Path ‘Test Plan Path

Set nowTest = QCUtil.CurrentTestSet ‘Current Testset

DSet nowTest = QCUtil.CurrentRun ‘Current TestsetInstance

**Reverse Integer:**

While (intNumtoReverse/10) > 0  
 varTemp = (varTemp\*10) + (intNumtoReverse Mod 10)  
 intNumtoReverse = Int(intNumtoReverse/10)  
Wend