UFT important stuff:

# QC connection:

#### Through QTP:

**Set** qtApp = **CreateObject**("QuickTest.Application")  
  
qtApp.TDConnection.Connect "http://app-hpqc-prd-03.ca.boeing.com:8080/qcbin/","BIT\_WIRS","WIRS","2777903","Password@123",**False**  
**If** qtApp.TDConnection.IsConnected **Then**  
    **msgbox** "success"  
**else**  
    **msgbox** "failed"  
**End** **If**

#### Through VBS:

**set** tdc = **createobject**("TDApiOle80.TDConnection")  
tdc.InitConnectionEx "http://app-hpqc-prd-03.ca.boeing.com:8080/qcbin/" *'enter your qc url here till qcbin*  
tdc.login "2777903","Password@123"  
  
tdc.Connect "BIT\_WIRS","WIRS"  
**If** tdc.Connected **then**  
    **msgbox** "Success"  
    **Else**  
    **msgbox** "Failed"  
**End** **if**

**Adding a defect in QC using QCUtil Object:**

*‘Create instance of QCConnection*

*Set QCConnection = QCUtil.QCConnection*

*‘Create an instance of BugFactory*

*Set DefFactory = QCConnection.BugFactory*

*'Add a new defect*

*Set Bug = DefFactory.AddItem(Nothing)*

*‘Provide mandatory details for the defect*

*Bug.Status = “New”*

*Bug.Summary = “Module Detected new defect summary”*

*Bug.DetectedBy = “njoshi”*

*Bug.AssignedTo = “dev001”*

*Bug.Post*

*Set DefFactory = nothing*

*Set QCConnection = nothing.*

**2. Adding Attachment to QC**

*Dim ObjCurrentTest,ObjAttch*

*‘ We can add attachment to currentTest, current run, testset, and testsettest by using repective ‘properties   
Set ObjTest = QCUtil.CurrentTest.Attachments  
Set ObjAttachFile = ObjTest.AddItem(Null)  
ObjAttachFile.FileName = FileName ‘ Provide path of the file that needs to be attached to QC   
ObjAttachFile.Type = 1  
ObjAttachFile.Post  
ObjAttachFile.Refresh*

**3. Validating If QC is connected properly:**

*if QCUtil.IsConnected then*

*msgbox “QC is connected”*

*Else*

*Msgbox “QC is not connected”*

*EndIf*

**4. Connecting to QC through QTP**

*Set qtApp = CreateObject ("QuickTest.Application")*

*If  qtApp.launched <> True then*

*qtApp.Launch*

*End If*

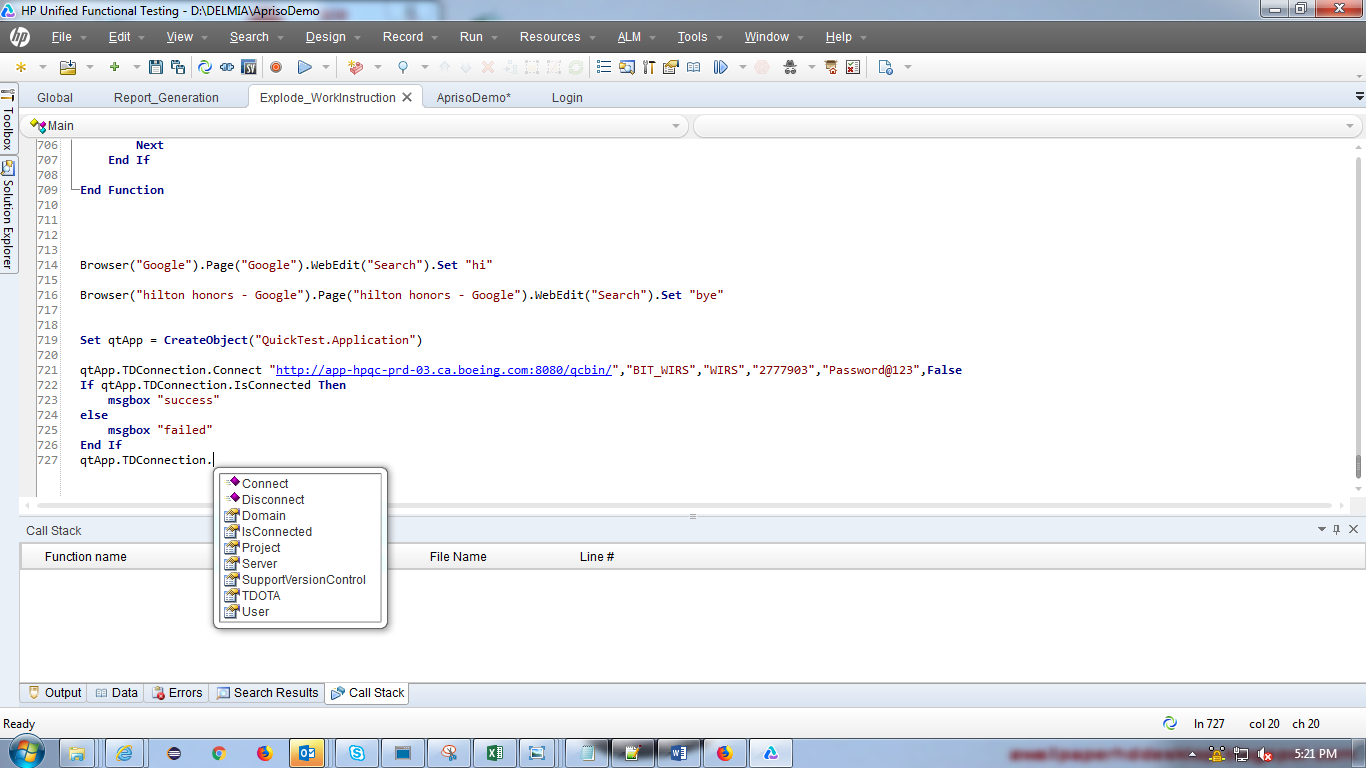
*qtApp.Visible = "true"*

*If Not qtApp.TDConnection.IsConnected Then*

*qtApp.TDConnection.Connect QCurl, DomainName, ProjectName, UserName, Password, False*

*End If*

QC connection:



qtApp.TDConnection.connect 🡪Connects UFT to ALM projet

qtApp.TDConnection.Disconnect 🡪 Disconnects ALM from UFT

qtApp.TDConnection.Domain 🡪 Displayed the domain to which its connected.

qtApp.TDConnection.IsConnected 🡪 Assures the connectivity

qtApp.TDConnection.Project 🡪 Displayed the project to which it is connected

qtApp.TDConnection.Server 🡪 Displays the server to which it is connected

qtApp.TDConnection. SupportVersionControl 🡪 Indiactes whether the UFT connected to ALM supports or not.

qtApp.TDConnection.TDOTA 🡪 Returns the TDOTA object, exposes the full interaction with ALM

qtApp.TDConnection.User 🡪 Displays the user connected to UFT

Results updater:

**Dim** qtApp *'As QuickTest.Application ' Declare the Application object variable*  
**Dim** qtUpdateRunOptions *'As QuickTest.UpdateRunOptions ' Declare an Update Run Options object variable*  
**Dim** qtRunResultsOptions *'As QuickTest.RunResultsOptions ' Declare a Run Results Options object variable*  
**Dim** blsSupportsVerCtrl *' Declare a flag for indicating version control support "Subject\Temp\Ana\DTM\_testing\" 01\_Editor\_launch\_max*  
**Dim** ArgObj, var1, var2  
var2 = "Subject\Temp\Ana\DTM\_testing\"  
**Set** ArgObj = WScript.Arguments  
  
*'First parameter*  
var1 = ArgObj(0)  
var2 = ArgObj(1)  
**Set** qtApp = **CreateObject**("QuickTest.Application") *' Create the Application object*  
qtApp.Launch *' Start QuickTest*  
qtApp.Visible = **True** *' Make the QuickTest application visible*  
  
*' Make changes in a test on Quality Center with version control*  
qtApp.TDConnection.Connect "QCurl", \_  
"domain", "prj", "userid", "pwd", **False** *' Connect to Quality Center*  
  
**If** qtApp.TDConnection.IsConnected **Then** *' If connection is successful*  
blsSupportsVerCtrl = qtApp.TDConnection.SupportVersionControl *' Check whether the project supports vervion control*  
  
*'Open the test*  
qtApp.Open "[QualityCenter] " &var1 &"\" &var2, **False** *' Open the test*  
  
**If** blsSupportsVerCtrl **Then** *' If the project supports version control*  
qtApp.Test.CheckOut *' Check out the test*  
**End** **If**  
  
*' Prepare the UpdateRunOptions object*  
**Set** qtUpdateRunOptions = **CreateObject**("QuickTest.UpdateRunOptions") *' Create the Update Run Options object*  
*' Set the Update Run options: update the Active Screen and test object descriptions. Do not update checkpoint values*  
qtUpdateRunOptions.UpdateActiveScreen = **True**  
qtUpdateRunOptions.UpdateCheckpoints = **False**  
qtUpdateRunOptions.UpdateTestObjectDescriptions = **True**  
  
*' Prepare the RunResultsOptions object*  
**Set** qtRunResultsOptions = **CreateObject**("QuickTest.RunResultsOptions") *' Create the Run Results Options object*  
*' qtRunResultsOptions.ResultsLocation = ""' Set a temporary results location*  
  
*'Update the test*  
qtApp.Test.UpdateRun qtUpdateRunOptions, qtRunResultsOptions *' Run the test in Update Run mode*  
qtApp.Test.**Description** = qtApp.Test.**Description** & **vbNewLine** & \_  
"Updated: " & **Now** *' Document the update in the test's description (Test Settings > Properties tab)*  
  
qtApp.Test.Save *' Save the test*  
  
**If** blsSupportsVerCtrl **And** qtApp.Test.VerCtrlStatus = "CheckedOut" **Then** *' If the test is checked out*  
qtApp.Test.CheckIn *' Check it in*  
**End** **If**  
qtApp.Test.Close *' Close the test*  
qtApp.TDConnection.Disconnect *' Disconnect from Quality Center*  
**Else**  
**MsgBox** "Cannot connect to Quality Center" *' If connection is not successful, display an error message.*  
**End** **If**  
  
qtApp.Quit *' Exit QuickTest*  
**Set** qtUpdateRunOptions = **Nothing** *' Release the Update Run Options object*  
**Set** qtRunResultsOptions = **Nothing** *' Release the Run Results Options object*  
**Set** qtApp = **Nothing** *' Release the Application object*  
**Set** ArgObj = **Nothing**  
*'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

# Excel Clean up:

**If**  Window("regexpwndtitle:=Excel","Index:=0").Exist (2) **then**   
             SystemUtil.CloseProcessByName("EXCEL.EXE")  
            **If** Dialog("text:=Microsoft Excel","Index:=0").WinButton("regexpwndtitle:=&Yes").Exist(2) **then**  
                Dialog("text:=Microsoft Excel","Index:=0").WinButton("regexpwndtitle:=&Yes").Click  
            **End** **If**   
            **If** Dialog("text:=Microsoft Office Excel","Index:=0").Exist (2) **Then**  
                **while** Dialog("text:=Microsoft Office Excel","Index:=0").Exist (2)  
                Dialog("text:=Microsoft Office Excel","Index:=0").WinCheckBox("text:=&Recover my work.\*").**Set** "OFF"  
                Dialog("text:=Microsoft Office Excel","Index:=0").WinButton("text:=&Don.\*").Click  
                **wend**   
            **End** **If**  
        **End** **If**

# **Event Mode:**

Setting.webpackage(“ReplayType”) = 2 ‘ 2 = Mouse 1 = Event

# Application invoking:

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

SystemUtil.Run "C://Program Files/Internet Explorer/IEXPLORE.EXE"

InvokeApplication("Full URL as Parameter")

InvokeApplication "C://Program Files/Internet Explorer/IEXPLORE.EXE http://www.yahoo.com"

#### VBS to invoke:

1. Dim oShellSet oShell = CreateObject ("Wscript.shell")'
2. 'Example 1 - run a batch file:
3. oShell.run "F://jdk1.3.1/demo/jfc/SwingSet2.bat"
4. 'Example 2 - run a Java jar file:
5. oShell.run "java -jar F://jdk1.3.1/demo/jfc/SwingSet2/SwingSet2.jar"
6. 'Example 3 - launch Internet Explorer:
7. oShell.Run Chr(34) & "C://Program Files/Internet Explorer/IEXPLORE.EXE" & Chr(34)
8. Set oShell = Nothing

#### Through windows:

If nothing works out you might try this

Use the Start -> Run dialog of Windows.

1. Add the Windows Start button to the Object Repository using the “Add Objects” button in Object Repository dialog.
2. Open the Run dialog (Start -> Run), and learn the “Open” edit field and the “OK” button into the Object Repository.
3. Switch to the Expert View, and manually add the lines to open the Run dialog.  
   Example:

Window("Window").WinButton("Button").ClickWindow("Window").Type("R")

1. Manually enter the lines to enter the information to launch the application, and click the “OK” button of the Run dialog.

**Example:**

Dialog("Run").WinEdit("Open:").Type "C://Windows/System32/notepad.exe"

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

#### **WebUtil** Object

In [UFT 14.01](https://www.learnqtp.com/uft-14-01-enhancements/) update, HPE introduced two new methods for WebUtil Object. **LaunchBrowser** and**LaunchMobileBrowserWithID**

Using **LaunchBrowser** method you can launch applications across desktop and mobile devices. The syntax for **LaunchBrowser** is

WebUtil.LaunchBrowser Browser, [device\_model, device\_manufacturer, device\_ostype, device\_osversion]

where:

* **Browser:** Name of browser to be launched. Ex: **CHROME, FIREFOX, IE**
* **device\_model:** The model of the selected device.
* **device\_manufacturer:**The name of manufacturer of the selected device.
* **device\_ostype:** The OS running on the device.
* **device\_osversion:**The OS version running on the selected device.

Parameters in square brackets are optional and they are meant to be used for mobile browsers only.

**Example:**

The following example launches the Chrome browser on an iOS Apple 5s device.

WebUtil.LaunchBrowser "MOBILE\_CHROME", "Apple\_5s", "Apple", "IOS", "10.1.3"

Using **LaunchMobileBrowserWithID**method you can launch a mobile browser on an iOS or an Android device using the Mobile Center’s device ID.

WebUtil.LaunchMobileBrowserWithID Browser, device\_ostype, device\_id

where

* **Browser:** Name of browser to be launched. Ex: **MOBILE\_HPWEB, MOBILE\_CHROME, MOBILE\_SAFARI**
* **device\_ostype:** The OS running on the device.
* **device\_id:** The id assigned to the device by Mobile Center.

**Example:**

WebUtil.LaunchMobileBrowserWithID "MOBILE\_CHROME", "IOS", "02"

#### Through object:

**Function** fn\_LaunchBrowser(URL)  
    **Set** objBro = **CreateObject**("internetexplorer.Application")  
    objBro.Visible = **True**  
    objBro.Navigate URL  
    **Set** objBro = **Nothing**  
**End** **Function**

# Excel handling:

**Function** DataSheetCellValue(lsSheetName, lsColumnName, lnRow, lsValue, lsOption)  
  
    DataSheetCellValue = ""  
  
*'Verifies the row is not null ot 0 (zero)*  
    **If** lnRow<=0 **Then**  
        **Exit** **Function**  
    **End** **If**  
      
    **On** **Error** **Resume** **Next**  
    **Err**.**Number** = 0  
*'Focusses the row in the specified sheet  and gets the value*  
    **Set** loSheet = DataTable.GetSheet(lsSheetName)  
    loSheet.SetCurrentRow(lnRow)  
    **Select** **Case** **LCase**(lsOption)  
        **Case** "get"  
            DataSheetCellValue = loSheet.GetParameter(lsColumnName).Value  
        **Case** "set"  
            loSheet.GetParameter(lsColumnName).Value = lsValue  
        **Case** **Else**  
            Reporter.ReportEvent micWarning, "Set or Get the value in specified sheet", "There is no option like "&lsOption  
    **End** **Select**  
  
    **If** **Err**.**Number**<>0 **Then**  
        Reporter.ReportEvent micWarning, "Set/Get the value in the '"&lsSheetName& "' sheet", "Either '"&lsSheetName&"' sheet or '"&lsColumnName&"' Column does not exist"  
    **End** **If**  
    **On** **Error** **Goto** 0  
      
  
**End** **Function**

# Customizing report:

Reporter.Filter= ***value***  
The ***value*** contains one of the following built-in variables.

1-rfEnableAll- Report all steps. **The default setting**

2-rfEnableErrorsAndWarnings- Only report error(failed)and warning steps.

3-rfEnableerrorsOnly- Only report error steps.

4-rfDisableAll- Does not report any steps.

*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'Function / Sub Name        :        VerifyErrorDialog*  
*'Description            :        Verifies and cllicks on OK or Yes buttons if any error dialog is opened in the specified browser*  
*'                :        If Security dialog exists clicks on Yesbutton*  
*'Input Arguments        :        oBrowser - Browser to check on which Dialog is being opened*  
*'Return                :        True - if exists / False*  
*'Written By            :        ChandraSekhar N         -        10/11/2009*  
*'MOdified By     :        ChandraSekhar N            -        23/02/2010    -    to capture the dialog title and te message in the dialog*  
*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
**Function** VerifyErrorDialog(oBrowser)   
  
    VerifyErrorDialog = **False**  
    lsResults = Reporter.**Filter**  
    *'For minimizing the reports*  
    Reporter.**Filter** = rfDisableAll  
  
*'Creates the Dialog Description*  
    **Set** oDlg = **Description**.Create()  
    oDlg("nativeclass").Value = "#32770"  
    oDlg("Class Name").Value = "Dialog"  
  
  
*'Verifies whether the Browser object exist in the Desktop*  
    **If** **NOT** oBrowser.Exist(1) **Then**  
        Reporter.**Filter** = lsResults  
        **Exit** **Function**  
    **End** **If**  
  
    lnMainCnt = 1  
    **Do**  
  
      
    *'Verifies whether the Dialog object opened in the Browser*  
        **If** **NOT** oBrowser.Dialog(oDlg).Exist(1) **Then**  
            Reporter.**Filter** = lsResults  
            **Exit** **Function**  
        **End** **If**  
      
        VerifyErrorDialog = **False**  
      
    *'Initializes the button text based on the Dialog title*  
        lsTitle = oBrowser.Dialog(oDlg).GetROProperty("text")  
      
    *'Modified by Chandra on 09/02/2010 to handle the unwanted popups from app*  
        **Set** oBtn = **Description**.Create()  
        oBtn("micclass").Value = "WinButton"  
        oBtn("text").RegularExpression = **True**  
        oBtn("text").Value = "(.\*Yes)|(.\*OK)"  
      
        **Set** oBtns = oBrowser.Dialog(oDlg).ChildObjects(oBtn)  
      
        **If** oBtns.Count = 0 **Then**  
            **Select** **Case** **LCase**(lsTitle)  
                **Case** "security alert","security"  
                    lsBtnText = "(\&|.)Yes"  
                **Case** **Else**  
                    lsBtnText = ".\*OK"  
                    *'lsBtnText = "OK)"*  
            **End** **Select**  
        **ElseIf** oBtns.Count = 1 **Then**  
            lsBtnText = oBtns(0).GetROProperty("text")  
            **Wait** 0,500  
        **End** **If**  
          
    *'Creates the Button Description based on the disalog title*  
        **Set** oButton = **Description**.Create()  
        oButton("text").RegularExpression = **True**  
        oButton("text").Value = lsBtnText  
  
        **Set** oStatic = **Description**.Create()  
        oStatic("micclass").Value = "Static"  
  
        **Set** oStatics = oBrowser.Dialog(oDlg).ChildObjects(oStatic)  
  
        **For** iLoop = 0 **To** oStatics.Count-1  
            **If** iLoop = 0 **Then**  
                lsText = oStatics(iLoop).GetROProperty("text")  
            **Else**  
                lsText = lsText & **VBCRLF** & oStatics(iLoop).GetROProperty("text")  
            **End** **If**  
        **Next**  
          
        *'Repeats the Loop till the Dialog gets close in 3 attempts*  
        lnCnt = 1  
        **Do**   
            **If** oBrowser.Dialog(oDlg).WinButton(oButton).Exist(1) **Then**  
                oBrowser.Dialog(oDlg).Activate  
                **Wait** 0,200  
                oBrowser.Dialog(oDlg).WinButton(oButton).Click  
                **Wait** 0,500  
                VerifyErrorDialog = **True**  
            **End** **If**  
            lnCnt = lnCnt + 1  
        **Loop** **While** oBrowser.Dialog(oDlg).Exist(1) **AND** lnCnt<=3  
  
        lnMainCnt = lnMainCnt + 1  
  
    **Loop** **While** oBrowser.Dialog(oDlg).Exist(1) **AND** lnMainCnt<=3  
      
    Reporter.**Filter** = rfEnableAll  
    Reporter.ReportEvent 4, "Verify Message dialog opened", "Dialog with the following Title and text is opened" &**VBCRLF**& "Title  :  "&lsTitle&**VBCRLF**& "Text  :  "&lsText  
*'To Enable the Log*  
    Reporter.**Filter** = lsResults  
  
*'Verifies whether the error dialog still exists*  
    **If** oBrowser.Dialog(oDlg).Exist(1) **Then**  
        Reporter.ReportEvent micWarning, "Close the error dialog opened in the DWR browser", "Could not able to close the "& lsTitle& " error dialog"  
    **End** **If**  
  
**End** **Function**

# Wait till object loaded:

*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'Function / Sub Name        :        WaitTillObjLoaded*  
*'Description            :        Verifies and waits till the object is loaded*  
*'Input Arguments        :        Obj     :    To be looked whether the object is exist and is loaded or not in the specified time*  
*'                :        lnTime    :    Time to wait to verify the object is exist and loaded*  
*'Return                :        True - if exists or  loaded / False*  
*'Written By            :        ChandraSekhar N         -        11/11/2009*  
*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
**Function** WaitTillObjLoaded(Obj, lnWaitTime)  
  
    **Dim** lnCnt, lsState, lnStartTime  
      
    lnStartTime = **Time**()  
      
    lsResults = Reporter.**Filter**  
    *'Disables the log*  
    Reporter.**Filter** = rfDisableAll  
      
    WaitTillObjLoaded = **False**  
  
    **If** **NOT** IsObjectExist(Obj,lnWaitTime) **Then**   
  
  **If** Browser("micclass:=Browser","title:=500 Internal.\*").Exist(0.1) **OR** Browser("micclass:=Browser","name:=500 Internal.\*").Exist(0.1)**Then**  
            Reporter.ReportEvent micWarning, "Verify that Error page is opned", "Redirected to error page"  
        **ElseIf** Browser("micclass:=Browser","title:=\*.Error.\*").Exist(0.1) **OR** Browser("micclass:=Browser","name:=.\*Error.\*").Exist(0.1)**Then**  
   Reporter.ReportEvent micWarning, "Verify that Error page is opned", "Redirected to Server error page"  
  **End** **If**  
  *'Enables the log*  
  *'''Call CaptureScreenShot(Browser("creationtime:=0").GetROProperty("hwnd"))*  
  Reporter.**Filter** = lsResults  
        **Exit** **Function**  
 **End** **If**  
      
 **On** **Error** **Resume** **Next**  
 lsState = ""  
    lnCnt = 1  
 **Do**  
    *'Gets the Object State*  
  lsState = Obj.Object.readystate  
  
  **If** **IsEmpty**(lsState) **Then**  
   WaitTillObjLoaded = **True**  
   Reporter.**Filter** = lsResults  
   **Exit** **Function**  
  **End** **If**  
  
  *'Converts the Object state from String to Numeric*  
  **Select** **Case** **LCase**(lsState)  
   **Case** "uninitialized"  
    lsState = 0  
   **Case** "loading"  
    lsState = 1  
   **Case** "loaded"  
    lsState = 2  
   **Case** "interactive"  
    lsState = 3  
   **Case** "complete"  
    lsState = 4  
  **End** **Select**  
  
*'   If State is >2 comes out*  
  **If** lsState>=2 **Then**  
   WaitTillObjLoaded = **True**  
   **Exit** **Do**  
  **End** **If**  
  lnEndTime = **Time**()  
  lnTotTime = **DateDiff**("s",lnStartTime, lnEndTime)  
 **Loop** **Until** lsState>=2 **AND** lnTotTime>=lnWaitTime  
  
*'Enables the log*  
Reporter.**Filter** = lsResults  
  
**End** **Function**

# DB Connection:

**Function** Get\_Db\_Value(SQL\_Statement,lsDBUserId,lsPassword,lsServerName)  
     **Dim** rs,conn,constr  
      **On** **Error** **Resume** **Next**  
    **Set** conn = **CreateObject**("ADODB.Connection")  
                constr="Driver={Microsoft ODBC for Oracle};Server="&lsServerName&";Uid="&lsDBUserId &";Pwd="&lsPassword&";"   
  
                conn.Open constr  
                Conn.CursorLocation = 2  
                  
                **Set** rs = **CreateObject**("ADODB.Recordset")  
                rs.CursorType = 3  
                rs.Open SQL\_Statement, conn  
  
                **If** rs.RecordCount < 1 **Then**          *' if there is no record in the table it will return a false string 'SERVICE\_NAME*  
                                Get\_Db\_Value= ""  
                **Else**  
                                Get\_Db\_Value= rs.Fields(0).Value  
                **End** **If**  
  
                **Set** rs = **Nothing**  
                **Set** conn=**Nothing**  
                **Set** constr=**Nothing**  
**End** **Function**        
  
*''\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
**Function** Get\_Db\_Values(SQL\_Statement,lsDBUserId,lsPassword,lsServerName)  
  *'  Dim rs,conn,constr*  
    **On** **Error** **Resume** **Next**  
    **Set** conn = **CreateObject**("ADODB.Connection")  
    constr="Driver={Microsoft ODBC for Oracle};Server="&lsServerName&";Uid="&lsDBUserId &";Pwd="& lsPassword &";"   
    conn.Open constr  
    Conn.CursorLocation = 2  
    **Set** rs = **CreateObject**("ADODB.Recordset")  
    rs.CursorType = 3      
    rs.Open SQL\_Statement, conn  
  
    **If** rs.RecordCount < 1 **Then**     *' if there is no record in the table it will return a false string*  
        Get\_Db\_Values= ""  
    **Else**  
        Recordset = rs.getRows()  
        Get\_Db\_Values= Recordset  
    **End** **If**  
  
    rs.Close  
    conn.close  
    **Set** rs = **Nothing**  
    **Set** conn=**Nothing**  
    **Set** constr=**Nothing**  
**End** **Function**

#### Update DB:

*'--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'Function / Sub Name                    :Update\_DB*  
*'Description                                        :                               To Update Values in the Database*  
*'Input Arguments                            :               SQL\_Statement, DB userName,Password,ServerName*  
*'Return                                                 :*  
*'Written By                                         :                               Chandrasekhar*  
*'--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
**Function** Update\_DB(SQL\_Statement,lsDBUserId, lsPassword, lsServerName)  
    **Err**.**Clear**  
    **Err**.**Number**=0  
    **On** **Error** **Resume** **Next**   
    **Dim** RecSet,ConStr  
    Update\_DB = ""  
    **Set** ConObj = **CreateObject**("ADODB.Connection")  
    constr="Driver={Microsoft ODBC for Oracle};Server="&lsServerName&";Uid="&lsDBUserId&";Pwd="&lsPassword&";"   
    ConObj.Open ConStr   
    ConObj.**Execute** SQL\_Statement  
    ConObj.**Execute** "Commit"  
      
    **If** **Err**.**Number**<>0 **Then** Update\_DB = **Err**.**Description**  
      
    **Err**.**Clear**  
    **Err**.**Number**=0  
    ConObj.Close  
    **Set** conn=**Nothing**  
**End** **Function**

# SendKeys:

**Function** SendKeys(lsKeys)  
    **On** **Error** **Resume** **Next**  
    **Set** Wsh = **CreateObject**("WScript.Shell")  
    Wsh.sendkeys lskeys  
    **Wait** 0,500  
    **On** **Error** **Goto** 0  
    **Set** Wsh = **Nothing**  
**End** **Function**

# String generator:

*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'Function / Sub Name        :        StringGenerator*  
*'Description            :        Generate Specified length of  string*  
*'Input Arguments        :        String Length*  
*'Return                :        String*  
*'Written By            :        Kingston Stephen         -        01/13/2011*  
*'-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
**Function** StringGenerator( **ByVal** lnstrLen,lsFormat )   
   **On** **Error** **Resume** **Next** :**Err**.**Number**=0  
    **Dim** lsStr  
    **Const** ALPHA = "abcdefghijklmno0111213141516171819202122232pqrstuvwxyz0133456789abcdefghijklmno0111213141516171819202122232pqrstuvwxyz0133456789abcdefghijklm0111213141516171819202122232nopqrstuvwxyz0133456789abcdefghijklmnopqrstuvwxyz0133456789"  
    **Const** NUMBERS = "1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768969707172"  
    **Const** LETTERS = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"  
    **Select** **Case** **LCase**(lsFormat)  
        **Case** "char"  
            **For** i = 1 **to** lnstrLen  
                lsStr = lsStr & **Mid**( LETTERS, RandomNumber( 1, **Len**( LETTERS ) ), 1 )  
            **Next**  
        **Case** "num"  
            **For** i = 1 **to** lnstrLen  
                lsStr = lsStr & **Mid**( NUMBERS, RandomNumber( 1, **Len**( NUMBERS ) ), 1 )  
            **Next**  
        **Case** "alpha"  
            **For** i = 1 **to** lnstrLen  
                lsStr = lsStr & **Mid**( ALPHA, RandomNumber( 1, **Len**( ALPHA ) ), 1 )  
            **Next**  
    **End** **Select**  
      
    StringGenerator = lsStr  
    **If** **Err**.**Number**<>0 **Then**  
        Browser("CreationTime:=0").Page("micClass:=Page").CaptureBitmap "StringGenerator.png",**True**  
        Reporter.ReportEvent 3,"Function : StringGenerator",**Err**.**Number** &"  "&**Err**.**Description**,"StringGenerator.png"  
    **End** **If**  
**End** **Function**

# Adding the OR at runtime:

*'''-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'''Function / Sub Name        :        fn\_Add\_OR*  
*'''Description            :        Verifies whether the OR is added or not at runtime*  
*'''Input Arguments        :        ls\_TSRPath*  
*''''Return Value             :*  
*'''Written By            :        kalyan        -        16/6/2011*  
*'''''modified By         :*  
*'''-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
**Function** fn\_Add\_OR(ls\_TSRPath)  
   **Dim** lsPreOR\_cnt,lsPostOR\_cnt  
   **If** **Len**(ls\_TSRPath) = 0  **Then**  
        Reporter.ReportEvent micWarning,"Verifying the shared OR path is provided or not","Shared OR path is not provided "  
        ExitAction(micWarning)  
   **End** **If**  
    lsPreOR\_cnt = RepositoriesCollection.Count  
*''    ls\_TSRPath = TestArgs("OR")*  
    RepositoriesCollection.Add ls\_TSRPath  
    lsPostOR\_cnt = RepositoriesCollection.Count  
    **If** **NOT** lsPostOR\_cnt >lsPreOR\_cnt  **Then**  
        Reporter.ReportEvent micWarning,"Verifying the shared OR is added or not","Shared OR is not added"  
        ExitAction(micWarning)  
    **End** **If**    
**End** **Function**

# Function is running:

*'------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
*'Function / Sub Name        :        fn\_IsRunning*  
*'Description            :        Function to get tcheck wether the process is alive*  
*'Input Arguments        :         Process name in capital letters*  
*'Return                :        True/False*  
*'Written By            :        Sathish Venkatesan*  
*'''------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*  
  
**Function** fn\_IsRunning(lsProcessNameInCapitalLetters)  
    **Dim**  AllProcess, Process  
    fn\_IsRunning= **False**   
    lsProcessNameInCapitalLetters=**UCASE**(lsProcessNameInCapitalLetters)  
  
        **Set** AllProcess = **getobject**("winmgmts:")  
  
        *'32 Bit*  
        **If**  AllProcess.InstancesOf("Win32\_process").Count>0 **Then**          
            **For** **Each** Process **In** AllProcess.InstancesOf("Win32\_process")  
                **If** (**Instr** (**Ucase**(Process.Name),lsProcessNameInCapitalLetters) = 1) **Then**  
                        fn\_IsRunning = **True**    
                        **Set** AllProcess = **nothing**  
                        **Exit** **Function**  
                **End** **If**   
            **Next**   
  
        *'64 Bit*  
        **ElseIf** AllProcess.InstancesOf("Win64\_process").Count>0 **Then**          
            **For** **Each** Process **In** AllProcess.InstancesOf("Win64\_process")  
                **If** (**Instr** (**Ucase**(Process.Name),lsProcessNameInCapitalLetters) = 1) **Then**  
                        fn\_IsRunning = **True**    
                        **Set** AllProcess = **nothing**  
                        **Exit** **Function**  
                **End** **If**   
            **Next**  
        **End** **If**  
    **Set** AllProcess = **nothing**  
**End** **Function**

# Webtable handling:

#### Retrieving data from web table:

rCount=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").RowCount

For r=1 to rCount

cCount=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").ColumnCount(r)

For c=1 to cCount

cData=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").GetCellData(r,c)

msgbox cData

Next

Next

#### Accessing child objects from web table

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Set oWebEdit=Description.Create

oWebEdit("micclass").value="WebEdit"

set objList=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").ChildObjects(oWebEdit)

For objIndex=0 to objlist.count-1

msgbox objlist(objIndex).getroproperty("html id")

Next

#### Accessing child objects from web table cell

rCount=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").RowCount

For r=1 to rCount

cCount=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").ColumnCount(r)

For c=1 to cCount

oCount=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").ChildItemCount(r,c,"WebEdit")

For obj=0 to oCount-1

set cObject=Browser("Yahoo! Mail: The best").Page("Yahoo! Mail: The best").WebTable("Yahoo! ID:").ChildItem(r,c,"WebEdit",obj)

msgbox cObject.getROproperty("html id")

Next

Next

Next

Difference between **childobject**, **childitem**  and **Getcelldata**methods

**ChildObjects**method is to access total child objects from web table object using description object.

Syntax:- *object***.ChildObjects** **(**[*Description*]**)**

**ChildItem**method is to access child objects from a web table cell in web table object without using description object.

Syntax:-*object***.ChildItem** **(***Row*, *Column*, *MicClass*, *Index***)**

**Getcelldata**method is to retrieve data from a web table cell in web table object.

Syntax:-*object***.GetCellData** **(***Row*, *Column***)**

# Datatable handling:

|  |  |  |
| --- | --- | --- |
| Datatable | DtSheet | DtParameter |
| **Methods**  AddSheet  DeleteSheet  Export  ExportSheet  GetCurrentRow  GetRowCount  GetSheet  GetSheetCount  Import  ImportSheet  SetCurrentRow  SetNextRow  SetPrevRow  **Properties**  GlobalSheet  LocalSheet  RawValue  Value | **Methods**  AddParameter  DeleteParameter  GetCurrentRow  GetParameter  GetParameterCount  GetRowCount  SetCurrentRow  SetNextRow  SetPrevRow  **Properties**  Name | **Properties**  Name  RawValue  Value  ValueByRow |

These are all the methods and properties have to use for doing any operation in QTP.

|  |  |
| --- | --- |
| Windows | QTP |
| 1. Create an Excel Sheet  2. Add Column to the sheet  3. Add values to the columns | 1. Datatable.Addsheet  2. AddParameter  3. Value & valuebyrow |

**Script Ex:- Adding Sheet**

Set dtSheet=Datatable.AddSheet(“Demo”) ‘ Creating Sheet

Set dtEmpid=dtSheet.Addparameter(“EmpId”,””) ‘ Creating Columns

Set dtEmpName=dtSheet.Addparameter(“EmpName”,””)

Set dtEmpSal=dtSheet.Addparameter(“Sal”,””)

For row=1 to 5

dtSheet.setcurrentrow(row)

dtEmpid.value=row ‘ Adding values to the Specific Rows

dtEmpName.value=”emp”&row

dtEmpSal.value= RandomNumber(3000,10000)

Next

Datatable.ExportSheet “C:\Test.xls”,”Demo”

Copy this code in to QTP and execute it. After executing you will find an excel sheet in C:\.

**Accessing data from the Sheet**

Suppose already you had a sheet in your drive and you want to access that data to use in QTP. In this case you need to import that external sheet into QTP and then you have to use getsheet & getparameter methods to access the data.

Script Ex:- Accessing Data

Datatable.Addsheet(“demo”)

datatable.ImportSheet dtPath, **,** *SheetSource, “demo”*

Set dg=datatable.GetSheet(“demo”)

ColumnCount=dg.getparametercount

RowCount=dg.getrowcount

For Column =1 to ColumnCount

Cname=dg.getparameter(Column).name

Set pmg=dg.getparameter(Cname)

For Row =1 to RowCount

Val=pmg.valuebyrow(Row)

Msgbox (val)

next

Next

Script Ex:- Copy Odd Values from one sheet

Suppose if you have some values like this…. In this if you want to copy odd values, and then you need to follow this code.

Set dtSheet=datatable.GetSheet(Sheetname)

Set dtColumn=dtSheet.getparameter(“Numbers”)

RowCount=dtSheet.getrowcount

For Row =1 to RowCount

Val= dtColumn.valuebyrow(Row)

If Val Mod 2 <>0 then

Msgbox (Val)

End If

next

In this Script for Sheetname you have to provide the sheet name from which sheet you’re going to get the data.

**Excel Object Model**

Excel object Model is to automate the excel operations. Using QTP we can create sheets, But using Excel Object Model we can do full pledged operations what ever we are doing on normal excel sheet in windows.

Here is a sample Script to know how we can work with EOM

Set Excel=CreateObject("Excel.Application")

Set ExcelSheet = CreateObject("Excel.Sheet")

ExcelSheet.Application.Visible = True

For row=1 to 10

ExcelSheet.ActiveSheet.Cells(row,1).Value = "This is column A, row"&row

Next

ExcelSheet.SaveAs "C:\TEST.XLS"

ExcelSheet.Application.Quit

Set ExcelSheet = Nothing

To know more about Excel Object Model go to...

Open Excel->Help menu-> Microsoft Excel Help-> Table of Contents-> Microsoft Excel Visual Basic Reference->Microsoft Excel Object Model

Some Interesting Questions…

1. Copy Odd Rows from a sheet
2. Copy employees whose age is above 50
3. Get Diagonal values from a sheet
4. Copy data from one sheet to another sheet without changing column names
5. Eliminate duplicate rows in a column
6. sort the values in a column

Excel from Sudhakar**:**

**Create excel file and enter some data save it**

'###############################################

'Create excel file and enter some data save it

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Add New Workbook

Set workbooks=excel.Workbooks.Add()

'Set the value in First row first column

excel.Cells(1,1).value="testing"

'Save Work Book

workbooks.saveas"D:\excel.xls"

'Close Work Book

workbooks.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set workbooks=Nothing

Set excel=Nothing

**Reading Values from a Specific excel Sheet**

'###############################################

' Reading Values from a Specific excel Sheet

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open the Excel File

Set workbook=excel.Workbooks.Open("D:\excel.xls")

'Get the Control on Specific Sheet

Set worksheet1=excel.Worksheets.Item("Sheet1")

' Display the Values

Msgbox worksheet1.cells(1,1).value

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set worksheet1=Nothing

Set workbook=Nothing

Set excel=Nothing

**Deleting Rows from Excel Sheet**

'###############################################

' Deleting Rows from Excel Sheet

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open the Excel File

Set workbook=excel.Workbooks.Open("D:\excel.xls")

'Get the Control on Specific Sheet

Set worksheet1=excel.Worksheets.Item("Sheet1")

'Delete Row1

worksheet1.Rows("1:1").delete

'Save Excel

workbook.SaveAs("D:\excel.xls")

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set worksheet1=Nothing

Set workbook=Nothing

Set excel=Nothing

**Add and Delete ExcelSheet**

'###############################################

' Add and Delete ExcelSheet

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open Existing Excel File

Set workbook=excel.Workbooks.Open("D:\excel.xls")

'Add New Sheet

Set newsheet=workbook.sheets.Add

'Assign a Name

newsheet.name="raj"

'Delete Sheet

Set delsheet=workbook.Sheets("raj")

delsheet.delete

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set newsheet=Nothing

Set delsheet=Nothing

Set workbook=Nothing

Set excel=Nothing

**Copy an Excel Sheet of one Excel File to another Excel File**

'###############################################

' Copy an Excel Sheet of one Excel File to another Excel File

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open First Excel File

Set workbook1=excel.Workbooks.Open("D:\excel1.xls")

'Open Second Excel File

Set workbook2=excel.Workbooks.Open("D:\excel2.xls")

'Copy data from first excel file sheet

workbook1.Worksheets("raj").usedrange.copy

'Paste Data to Second Excel File Sheet

workbook2.Worksheets("Sheet1").pastespecial

'Save Workbooks

workbook1.Save

workbook2.Save

'Close Workbooks

workbook1.Close

workbook2.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set workbook1=Nothing

Set workbook2=Nothing

Set excel=Nothing

**Comapre Two Excel Sheets Cell By Cell for a specific Range**

'###############################################

' Comapre Two Excel Sheets Cell By Cell for a specific Range

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open Excel File

Set workbook=excel.Workbooks.Open("D:\excel.xls")

'Get Control on First Sheet

Set sheet1=excel.Worksheets.Item("Sheet1")

'Get Control on Second Sheet

Set sheet2=excel.Worksheets.Item("Sheet2")

'Give the specific range for Comparision

CompareRangeStartRow=1

NoofRows2Compare=4

CompareRangeStartColumn=1

NoofColumns2Compare=4

'Loop through Rows

For r=CompareRangeStartRow to(CompareRangeStartRow+(NoofRows2Compare-1))

'Loop through columns

For c=CompareRangeStartColumn to(CompareRangeStartColumn+(NoofColumns2Compare-1))

'Get Value from the First Sheet

value1=Trim(sheet1.cells(r,c))

'Get Value from the Second Sheet

value2=Trim(sheet2.cells(r,c))

'Compare Values

If value1<>value2 Then

' If Values are not matched make the text with Red color

sheet2.cells(r,c).font.color=vbred

End If

Next

Next

'Save workbook

workbook.Save

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set sheet1=Nothing

Set sheet2=Nothing

Set workbook=Nothing

Set excel=Nothing

**Reading complete data from excel file**

'###############################################

' Reading complete data from excel file

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open Excel File

Set workbook=excel.Workbooks.Open("D:\excel.xls")

'Get Control on Sheet

Set worksheet=excel.Worksheets.Item("raj")

'Get the count of used columns

ColumnCount=worksheet.usedrange.columns.count

'Get the count of used Rows

RowCount=worksheet.usedrange.rows.count

'Get the Starting used Row and column

top=worksheet.usedrange.row

lft=worksheet.usedrange.column

'Get cell object to get the values cell by cell

Set cells=worksheet.cells

'Loop through Rows

For row=top to (RowCount-1)

rdata=""

'Loop through Columns

For col=lft to ColumnCount-1

'Get Cell Value

word=cells(row,col).value

'concatenate all row cell values into one variable

rdata=rdata&vbtab&word

Next

'Print complete Row Cell Values

print rdata

Next

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set worksheet=Nothing

Set workbook=Nothing

Set excel=Nothing

**Read complete data from an Excel Sheet content**

'###############################################

' Read complete data from an Excel Sheet content

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Open Excel File

Set workbook=excel.Workbooks.open("D:\excel.xlsx")

'Get Control on Sheet

Set worksheet=excel.Worksheets.Item("Sheet1")

'Get Used Row and Column Count

rc=worksheet.usedrange.rows.count

cc=worksheet.usedrange.columns.count

'Loop through Rows

For Row=1 to rc

'Loop through Columns

For Column=1 to cc

'Get Cell Data

RowData=RowData&worksheet.cells(Row,Column)&vbtab

Next

RowData=RowData&vbcrlf

Next

'Display complete Data

msgbox RowData

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set worksheet=Nothing

Set workbook=Nothing

Set excel=Nothing

**Assign Colours to Excel Sheet Cells, Rows**

'###############################################

' Assign Colours to Excel Sheet Cells, Rows

'###############################################

'Create Excel Object

Set excel=createobject("excel.application")

'Make it Visible

excel.Visible=True

'Add a New work book

Set workbook=excel.workbooks.add()

'Get the Excel Sheet

Set worksheet=excel.worksheets(1)

'Coloring Excell Sheet Rows

Set objrange=excel.activecell.entirerow

objrange.cells.interior.colorindex=37

'Coloring Excell Sheet Cell

worksheet.cells(2,1).interior.colorindex=36

'Save Excel

workbook.SaveAs("D:\excel.xls")

'Close Work Book

workbook.Close

'Quit from Excel Application

excel.Quit

'Release Variables

Set objrange=Nothing

Set worksheet=Nothing

Set workbook=Nothing

Set excel=Nothing

# XML Handling:

#### -:Working With XML:-

Xml is markup language for documents containing structured information. Xml stands for "Extensible markup language".

**Sample XML:**

<Environment>

<Variable>

<id>602345</id>

<name  title ="Testing Engineer"  compnay="xyz">vikash</name>

</Variable>

<Variable>

<id>602346</id>

<name  title ="Developer"  compnay="mnp">rahul</name>

</Variable>

</Environment>

**Key Elements:**

**Root node:**

The first node in an xml file is called the root node. A valid xml file can have only one root node. In above example, the **"<Environment>"**node is the root node.

**Child Nodes:**

Child nodes are the children of an enclosing parent node. In above example  the **"<Variable>"**node is a child node of the   **"<Environment>"**parent node**;**whilethe **"<id>"** and **"<name>"** nodes are children of the **"<Variable>"** node.

**Node values:**

Node value is the value assigned to a given node. A node may only have a value assignment if it  does not have any child nodes. Here the first **"<id>"**node has a value assignment of  "602345" and **"<name>"**node has a value assignment of "vikash".

**Attributes:**

A nodecan have attributes. In the first  **"<name>"** node which is  **"<name  title ="Testing Engineer"  compnay="xyz">"** here **"title"** and **"company"** are attributes and **"Testing Engineer"** and **"xyz"** are their attribute values respectively.

**QTP XML Objects:-**

QTP provides a rich set of utility XML objects, the most important of which are described as below

1. **XMLUtil:-**It's used to create or read an XML file
2. **XMLAttribute:-**Provides access to the name and value of attributes of a specified node
3. **XMLAttributesColl:-**Provides a collection of attributes
4. **XMLElement:-**Represent an XML Element. An XML element can have attributes, a value, and/or one or more children, all of XMLElement type
5. **XMLElementColl:-**Provides a collection of XML elements
6. **XMLItemColl:-**Provides a collection of XML Items. XML items are used for CDATASection and comments in the XML.
7. **XMLData:-**Representan XML Block. XML Data contains the complete XML file loaded using XMLUtil or from WebXML test object.

**Reading And Modifying XML:-**

1. **Copy an XML file and Save it to another XML file**

**Method 1:**

**'**create an empty XMLData object

Set xmlDocument = XMLUtil.CreateXML

'Initialize the object with XML from a file

xmlDocument .LoadFile "C:\source.xml"

'Save it to a specified destination file

xmlDocument .SaveFile "C:\destination.xml"

Set xmlDocument  = Nothing

**Method 2:**

'Createan XMLData object with file data in a single step

Set xmlDocument  = XMLUtil.CreateXMLfromFile("C:\source.xml")

xmlDocument .SaveFile "C:\destination.xml"

Set xmlDocument  = Nothing  
2. **How can we get an entire xml file into a string and then save it to another XML file using that string  ?**

'Createan XMLData object with file data in a single step

Set xmlDocument  = XMLUtil.CreateXMLfromFile("C:\source.xml")

sXMLData = xmlDocument.ToString

'Create a 2nd XMLData object using the string as the initializer

Set xmldoc = XMLUtil.CreateXML()

Xmldoc.Load sXMLData

Xmldoc.SaveFile "C:\destination.xml"

Set xmlDocument  = Nothing

Set xmldoc = Nothing  
3. **How to Create XML file at run time and save it to an XML file ?**

<Environment>

<Company>

<id> 215532</id>

<name>Vijen</name>

</Company>

<Company>

<id>378657</id>

<name>Vijendra</name>

</Company>

</Environment>

**Script:**

'Create an empty XMLData object

Set xmlDocument = XMLUtil.CreateXML

'Create a document with Environment as the root node

xmlDocument .CreateDocument "**Environment**"

'Get the root not element

'The Data type returned would be of type XMLElement

Set xmlRoot = xmlDocument .GetRootElement()

**'Add a new Company node**

Set newNode = xmlRoot.AddChildElementByName("Company", "")

'Add the id and name child nodes

newNode.AddChildElementByName "id", "215553"

newNode.AddChildElementByName "name", "Vijen"

**'Add a new Company node**

Set newNode = xmlRoot.AddChildElementByName("Company", "")

'Add the id and name child nodes

newNode.AddChildElementByName "id", "378657"

newNode.AddChildElementByName "name", "Vijendra"

xmlDocument.SaveFile "C:\destination.xml"

Set newNode = Nothing

Set xmlRoot = Nothing

Set xmlDocument = Nothing  
4. **Different Way to Read  XML file:-**

**XMFile Content: CompanyXML.xml**

  <Environment>

    <Company organization="Wipro" startDate="14/02/2011">

      <id> 215532</id>

      <name>Vijen</name>

    </Company>

    <Company organization="Deloitte" startDate="20/01/2014">

      <id>378657</id>

      <name>Vijendra</name>

    </Company>

  </Environment>  
A) **Reading XML file using Node or Element Name**

**Dim** fileName

**Dim** xmlDoc, xmlRoot

fileName = "C:\Users\VMalav\Documents\Vijen\My\_Work\AUTOMATION TOOL WORK\QTP WORK\XML\_Development\CompanyXML.xml"

**Set** xmlDoc = XMLUtil.CreateXMLFromFile(fileName)

**Set** xmlRoot = xmlDoc.GetRootElement()

*'Get all the company node*

**Set** oAllCompanyNodes = xmlRoot.ChildElementsByPath("Company")

*'Iterate All the Company nodes*

**For** iNode = 1 **To** oAllCompanyNodes.Count

**Set** companyNode = oAllCompanyNodes.Item(iNode)

    vId = companyNode.ChildElementsByPath("id").Item(1).Value

    vName = companyNode.ChildElementsByPath("name").Item(1).Value

    Print "Id:= "& vId & " Name:= "& vName

**Next**

**Output:**

Id:=  215532 Name:= Vijen

Id:= 378657 Name:= Vijendra  
B) **Reading XML file using ChildElements**

**Dim** fileName

**Dim** xmlDoc, xmlRoot

fileName = "C:\Users\VMalav\Documents\Vijen\My\_Work\AUTOMATION TOOL WORK\QTP WORK\XML\_Development\CompanyXML.xml"

**Set** xmlDoc = XMLUtil.CreateXMLFromFile(fileName)

**Set** xmlRoot = xmlDoc.GetRootElement()

*'Get all the company node*

**Set** oAllCompanyNodes = xmlRoot.ChildElements()

*'Iterate All the Company nodes*

**For** iCompanyNode = 1 **To** oAllCompanyNodes.Count

**Set** companyNode = oAllCompanyNodes.Item(iCompanyNode)

**Set** childCompanyNode = companyNode.ChildElements()

*'Iterate All the node inside Company node like id, name*

**For** iNode = 1 **To** childCompanyNode.Count

        vElementName  = childCompanyNode.Item(iNode).ElementName

        vElementValue = childCompanyNode.Item(iNode).Value

        Print "Element Name : "& vElementName &" Element Value : "& vElementValue

**Next**

**Next**

**Output:-**

Element Name : id Element Value :  215532

Element Name : name Element Value : Vijen

Element Name : id Element Value : 378657

Element Name : name Element Value : Vijendra

**5. Different Way to read attribute of specified node**

**A) Read Attribute by item method**

**Dim** fileName

**Dim** xmlDoc, xmlRoot

fileName = "C:\Users\VMalav\Documents\Vijen\My\_Work\AUTOMATION TOOL WORK\QTP WORK\XML\_Development\CompanyXML.xml"

**Set** xmlDoc = XMLUtil.CreateXMLFromFile(fileName)

**Set** xmlRoot = xmlDoc.GetRootElement()

*'Get all the company node*

**Set** oAllCompanyNodes = xmlRoot.ChildElementsByPath("Company")

*'Iterate All the Company nodes*

**For** iNode = 1 **To** oAllCompanyNodes.Count

**Set** companyNode = oAllCompanyNodes.Item(iNode)

*'Get all the attribute of company node*

**Set** attributeCol = companyNode.Attributes

*'Iterate all the attributes*

**For** iAttribute = 1 **To** attributeCol.Count

**Set** attribute = attributeCol.Item(iAttribute)

        attributeName = attribute.Name

        attributeValue = attribute.Value

        Print "Attribute Name:= "& attributeName & " || Attribute Value:= "& attributeValue

**Next**

**Next**

**Output:**

Attribute Name:= organization || Attribute Value:= Wipro

Attribute Name:= startDate || Attribute Value:= 14/02/2011

Attribute Name:= organization || Attribute Value:= Deloitte

Attribute Name:= startDate || Attribute Value:= 20/01/2014

**B) Read Attribute by attribute name**

**Dim** fileName

**Dim** xmlDoc, xmlRoot

fileName = "C:\Users\VMalav\Documents\Vijen\My\_Work\AUTOMATION TOOL WORK\QTP WORK\XML\_Development\CompanyXML.xml"

**Set** xmlDoc = XMLUtil.CreateXMLFromFile(fileName)

**Set** xmlRoot = xmlDoc.GetRootElement()

*'Get all the company node*

**Set** oAllCompanyNodes = xmlRoot.ChildElementsByPath("Company")

*'Iterate All the Company nodes*

**For** iNode = 1 **To** oAllCompanyNodes.Count

**Set** companyNode = oAllCompanyNodes.Item(iNode)

*'Get all the attribute of company node*

**Set** attributeCol = companyNode.Attributes

    vAttrOrganization = attributeCol.ItemByName("organization").Value

    vStartDate = attributeCol.ItemByName("startDate").Value

    Print "Organization:= "& vAttrOrganization & " || StartDate:= "& vStartDate

**Next**

**Output:**

Organization:= Wipro || StartDate:= 14/02/2011

Organization:= Deloitte || StartDate:= 20/01/2014

**6. Add attribute at run-time:-**

*' Add Attribute to the spcified node based on condition*

**If** vStartDate = "14/02/2011" **Then**

        companyNode.AddAttribute "endDate", "30/10/2013"

**else**

        companyNode.AddAttribute "endDate", "27/03/2015"

**End** **If**  
 **7. Remove Attribute at run time**

companyNode.RemoveAttribute("endDate")

**8. Add and Remove Comment to the specified node:-**  
 **A) Add Comment**

companyNode.AddComment("hello")

**Output:**

<Environment>

  <Company organization="Wipro" startDate="14/02/2011">

    <id> 215532</id>

    <name>Vijen</name>

    <!--hello-->

  </Company>

  <Company organization="Deloitte" startDate="20/01/2014">

    <id>378657</id>

    <name>Vijendra</name>

    <!--hello-->

  </Company>

</Environment>  
B) **Remove Comment**

companyNode.RemoveComment("hello")

**Output:**

<Environment>

  <Company organization="Wipro" startDate="14/02/2011">

    <id> 215532</id>

    <name>Vijen</name>

  </Company>

  <Company organization="Deloitte" startDate="20/01/2014">

    <id>378657</id>

    <name>Vijendra</name>

  </Company>

</Environment>  
8. **Update Node or Element value using "SetValue":-**

companyNode.ChildElementsByPath("name").Item(1).SetValue "Vijen Malav"

**Compare Two XML File**:-

Both xml file should have same node ordering.

' Load both xml

Set xmlDocument1 = XMLUtil.CreateXMLFromFile("D:\company1.xml")

Set xmlDocument2 = XMLUtil.CreateXMLFromFile("D:\company2.xml")

' use compare method for equality

bIsEqual = xmlDocument1 .Compare(xmlDocument1, )

If bIsEqual  = 1 Then

Print "both file are same"

Else

Print "both file are not same"

End if

**How to Use "XMLUtil Object" in external vbscript file:-**

We can use XMLUtil object using com object

Set XMLUtil = CreateObject("Mercury.XMLUtil")

# Text file handling:

#### To Write:

Set objFileToWrite = CreateObject("Scripting.FileSystemObject").OpenTextFile("C:\listfile.txt",2,true)

objFileToWrite.WriteLine(data)

objFileToWrite.Close

Set objFileToWrite = Nothing

#### OpenTextFile parameters:

<filename>, IOMode (1=Read,2=write,8=Append), Create (true,false), Format (-2=System Default,-1=Unicode,0=ASCII)

#### To Read the entire file:

Set objFileToRead = CreateObject("Scripting.FileSystemObject").OpenTextFile("C:\listfile.txt",1)

strFileText = objFileToRead.ReadAll()

objFileToRead.Close

Set objFileToRead = Nothing

#### To Read line by line:

Set objFileToRead = CreateObject("Scripting.FileSystemObject").OpenTextFile("C:\listfile.txt",1)

Dim strLine

do while not objFileToRead.AtEndOfStream

strLine = objFileToRead.ReadLine()

'Do something with the line

loop

objFileToRead.Close

Set objFileToRead = Nothing

#### Change a read-only file to a read-write file

Const ReadOnly = 1

Dim fso, file

Dim filePathStr

filePathStr = "C:\MyFile.txt"

Set fso = CreateObject( "Scripting.FileSystemObject" )

If fsoutil.FileExists( filePathStr ) Then

Set file = fso.GetFile( filePathStr )

If file.Attributes And ReadOnly Then

file.Attributes = file.Attributes Xor ReadOnly

End If

Else

Reporter.ReportEvent micWarning, "Change Attributes", "File -> " & filePathStr & " was not found."

End If

Begin by setting the value of the constant **ReadOnly** to 1. We then create an instance of the **FileSystemObject**, and use the **GetFile** method to bind to the file filePathStr.

The Read-only attribute is stored as part of a “bitmask” along with other file attributes such as Hidden (value of 2, which indicates whether the file is a hidden file) and System (value of 4, which indicates whether the file is a system file). In a bitmask individual attributes can be likened to switches, switches that can be either on or off. If the switch with value of 1 is on, then the file is read-only; if the switch with the value of 1 is off, then the file is read-write

So, how you determine whether one of these switches is on, or off? The simple answer is that you use the bitwise **And** operator. Notice the following odd-looking line in the script.

If file.Attributes And ReadOnly Then

This line is checking to see if the **ReadOnly** attribute (with a value of 1) is on.

What if we wanted to see if the file was a hidden file? Well, in that case, we’d use this line of code.

If file.Attributes And Hidden Then

Basically the **And** operator can be read like this: “If we’re looking at the **Attributes** of the file and of the **ReadOnly** switch is on, then this is a read-only file and this If statement is True. If the **ReadOnly** switch is off, then this If statement is False.”

So why are we checking to see if the file is read-only? Well, momentarily, we’re going to use the **Xor** operator to “flip” the switch; that’s what this line of code is for?

file.Attributes = file.Attributes Xor ReadOnly

In this example, **Xor** simply toggles the file from one state to the other. If the file is read-only, **Xor** switches it to read-write; if it’s read-write, **Xor** switches it to read-only. That’s why we first check to see if the file is already read-only. If it is, we want to flip the switch and make it read-write. If it’s already read-write, though, we obviously don’t want to flip the switch; after all, that would make the file read-only.

here’s a script which binds to the folder pathStr and turns all the read-only files in that folder into read-write files.

Option Explicit

Const ReadOnly = 1

Dim fso, file, files, folder

Dim pathStr

pathStr = "C:\MyFolder"

Set fso = CreateObject( "Scripting.FileSystemObject" )

Set folder = fso.GetFolder( pathStr )

Set files = folder.Files

For Each file In files

If file.Attributes And ReadOnly Then

file.Attributes = file.Attributes Xor ReadOnly

End If

Next

UFT questions:

* What is the most challenging technical difficulty they came across during test automation and how do they overcome it?
* Descriptive programming takes a heavy hit on performance. How do you handle dynamically changing objects?
* How do you iterate over a report which is presented in a tabular format?
* Write a program to convert a date time object to different locale. For example: var currentDateAndTime = Now(). Convert the currentDateAndTime to India and US locale (DD/MM/YYYY and MM/DD/YYYY formats)
* How do you compare two String objects?
* Do you use recovery scenarios? (I don’t use them. I prefer error handling in code which is fast and reliable)
* How do you handle time-out issues? For example, application takes time to process data. If there are 50 records, it would take 5 seconds. If there are 5000 records, it would take 100 seconds. How will you handle such situations?
* What would you do when the deadline is approaching and you haven’t completed the task?
* Write a sample test case as you did in your current project
* Write the content of a bug as you did in your current project. Ask how priority and severity are decided in your project
* A bug that you raised was rejected by dev team. How do you handle this situation?
* One of your team member is not productive. It’s affecting your deliverables and deadline. What would you do in such situation?

I may not ask only QTP related questions. I would ask about framework, estimation, process and behavioural questions. Hope this helps.

1. Explain to me the criteria for what types of tests should be automated and what should not.
2. Describe some different automation frameworks you have developed and why you chose different architectures.
3. I would then test the candidates proficiency in VBScript.
4. Lastly I would have the candidate explain each of the following and the role each plays in using UFT:
   1. Object Repository
   2. Function Libraries
   3. Action Libraries
5. Then, how parameters are passed between actions.
6. How to handle a non-standard object using a user-defined function and how to call it using the normal parent.object hierarchy.

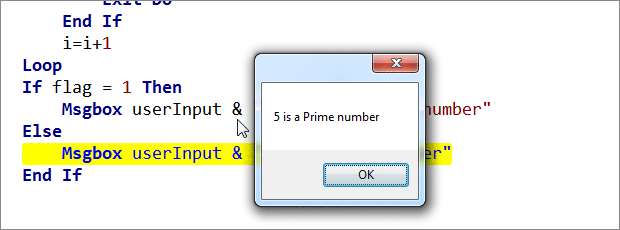
I might also ask them some questions about the various types of action calls and how they impact code reusability.

**Given Number is Prime or not?**

**Maths:** A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

**Logic:** Starting from 2 divide the given number till half of the number, if the remainder is zero then the number is not a prime number

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | userInput = Inputbox ("Enter any number")  i=2  Do While i < userInput/2      If userInput Mod i = 0 Then          flag = 1          Exit Do      End If      i=i+1  Loop  If flag = 1 Then      Msgbox userInput & " is not a Prime number"  Else      Msgbox userInput & " is a Prime number"  End If |

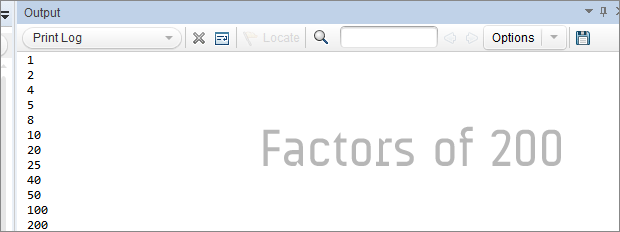
**Output:**  
[](https://i0.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Prime-number.png)

**2. Find the factors of a given number?**

**Maths:** The factors of a number are all those numbers that can divide evenly into the number with no remainder.

**Logic:** starting from 1 divide the given number till half of the number, if the remainder is zero it is a factor of the given number.

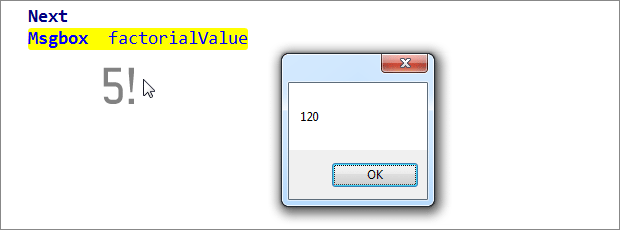
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | userInput = Inputbox ("Enter any number other than Prime number")  For i = 1 to userInput/2      If userInput Mod i = 0 Then          print i      End If  Next  print userInput |

**Output:**  
[](https://i0.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Factors.png)

**3. Find factorial of given number?**

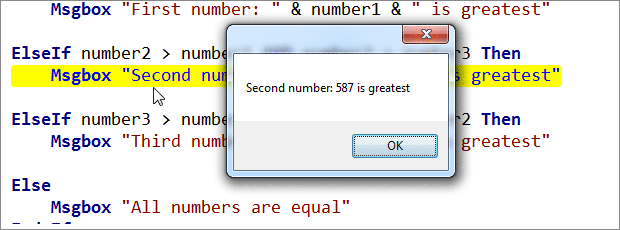
**Maths:** Factorial is the product of all the numbers from 1 to n, where n is the user specified number.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | factorialValue = 1  userInput = Inputbox ("Enter any number")  For i = 1 to userInput      factorialValue = factorialValue \* i  Next  Msgbox  factorialValue |

**Output:**  
[](https://i0.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Factorial.png)

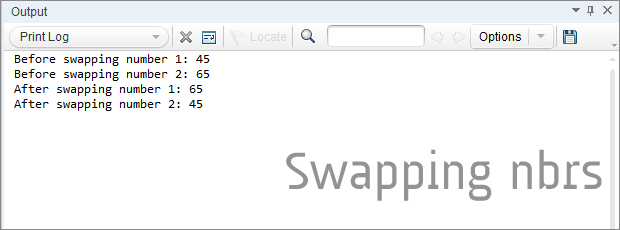
**4. Find greatest of three numbers?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | number1 = Inputbox ("First number: Enter any postive number")  number2 = Inputbox ("Second number: Enter any postive number")  number3 = Inputbox ("Third number: Enter any postive number")    If number1 > number2 AND number1 > number3 Then      Msgbox "First number: " & number1 & " is greatest"    ElseIf number2 > number1 AND number2 > number3 Then      Msgbox "Second number: " & number2 & " is greatest"    ElseIf number3 > number1 AND number3 > number2 Then      Msgbox "Third number: " & number3 & " is greatest"    Else      Msgbox "All numbers are equal"  End If |

**Output:**  
[](https://i2.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Greatest-of-Three-nbrs.png)

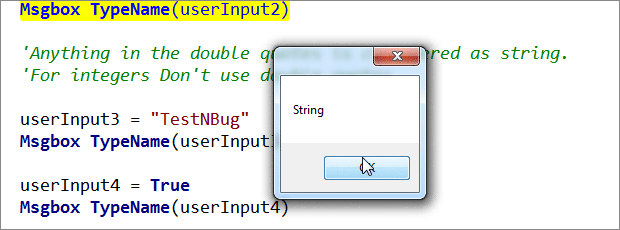
**5. Swap 2 numbers without a temporary variable**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | number1 = Inputbox ("First number: Enter any postive number")  number2 = Inputbox ("Second number: Enter any postive number")    Print "Before swapping number 1: " & number1  Print "Before swapping number 2: " & number2    number1 = number1 - number2  number2 = number1 + number2  number1 = number2 - number1    Print "After swapping number 1: " & number1  Print "After swapping number 2: " & number2 |

**Output:**  
[](https://i1.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Swapping-nbrs.png)

**6. Write a program to find sub datatype of a variable**

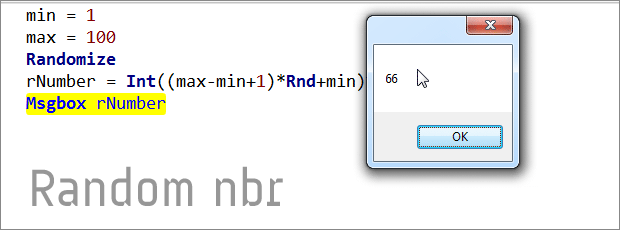
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | userInput1 = 44  Msgbox TypeName(userInput1)    userInput2 = "44"  Msgbox TypeName(userInput2)    'Anything in the double quotes is considered as string.  'For integers Don't use double quotes    userInput3 = "TestNBug"  Msgbox TypeName(userInput3)    userInput4 = True  Msgbox TypeName(userInput4) |

**Output:**  
[](https://i2.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Subdata-type.png)

**7. Write a program to generate Random Number**

We have a predefined formula for random number generation  
You need to enter the minimum and maximum limits.

|  |  |
| --- | --- |
| 1  2  3  4  5 | min = 1  max = 100  Randomize  rNumber = Int((max-min+1)\*Rnd+min)  Msgbox rNumber |

**Output:**  
[](https://i0.wp.com/www.testnbug.com/wp-content/uploads/2015/07/Random-nbr.png)

VBScript Program - Hello World [Link](http://qtp.blogspot.com/2010/11/vbscript-hello-world.html)  
VBScript Program - Get input text from user [Link](http://qtp.blogspot.com/2010/11/vbscript-input-text-from-user.html)  
VBScript Program - Square of a number [Link](http://qtp.blogspot.com/2010/11/vbscript-square-of-number.html)  
VBScript Program - Reversing a string [Link](http://qtp.blogspot.com/2010/11/vbscript-reversing-string.html)  
VBScript Program - Replacing substring with another in a string [Link](http://qtp.blogspot.com/2010/11/vbscript-replacing-string-with-another.html)  
VBScript Program - Getting positing of a substring in a string [Link](http://qtp.blogspot.com/2010/11/vbscript-position-of-string.html)  
VBScript Program - Extracting substring from string [Link](http://qtp.blogspot.com/2010/11/vbscript-get-characters-from-string.html)  
VBScript Program - Creating inverse triangle [Link](http://qtp.blogspot.com/2010/11/vbscript-inverse-triangle.html)  
VBScript Program - Palindrome [Link](http://qtp.blogspot.com/2010/11/vbscript-palindrome.html)  
VBScript Program - Appending text in a file [Link](http://qtp.blogspot.com/2010/11/vbscript-appends-text-to-file.html)  
VBScript Program - Reading a file [Link](http://qtp.blogspot.com/2010/11/vbscript-reading-file.html)  
VBScript Program - Reading lines from file [Link](http://qtp.blogspot.com/2010/11/vbscript-reads-lines-from-file.html)  
VBScript Program - Writing into file [Link](http://qtp.blogspot.com/2010/11/vbscript-write-line-into-file.html)  
VBScript Program - Addition function [Link](http://qtp.blogspot.com/2010/11/vbscript-addition-function.html)  
VBScript Program - Factorial function [Link](http://qtp.blogspot.com/2010/11/vbscript-factorial-function.html)  
VBScript Program - Prime or not [Link](http://qtp.blogspot.com/2010/11/vbscript-prime-number-or-not.html)  
VBScript Program - Fibonacci series [Link](http://qtp.blogspot.com/2010/11/vbscript-fibonacci-series.html)  
VBScript Program - Reading excel [Link](http://qtp.blogspot.com/2010/11/vbscript-reading-values-from-excel.html)  
VBScript Program - Writing to excel [Link](http://qtp.blogspot.com/2010/11/vbscript-writing-value-to-excel.html)  
VBScript Program - Counting words, spaces, characters etc. [Link](http://qtp.blogspot.com/2010/11/vbscript-counting-words.html)  
VBScript Program - Sum of digits of 5 digit number [Link](http://qtp.blogspot.com/2010/11/vbscript-sum-five-digit-number.html)  
VBScript Program - Sorting an array [Link](http://qtp.blogspot.com/2010/11/vbscript-sorting-array.html)  
VBScript Program - Understanding simple class [Link](http://qtp.blogspot.com/2010/11/vbscript-understanding-simple-class.html)

<http://qtp-test.blogspot.com/p/common-functions-in-qtp.html>

<https://www.learnqtp.com/forums/Thread-QTP-connection-to-QC>

<https://community.softwaregrp.com/t5/Quality-Center-ALM-User/OTA-Script-to-fetch-test-lab-results-data/td-p/941816>