**How to select specific web checkbox in QTP?**

Sometimes there are webpages having multiple check boxes. You have to select specific check box.  
  
We can solve this problem by 3 ways as mentioned below in QTP.

1. Using index property
2. Using Description object
3. Using HTML DOM

**Selecting the web check box using index -**

if Browser("myb").page("myp").webcheckbox("index:=2").getROProperty("checked") = false then  
     Browser("myb").page("myp").webcheckbox("index:=2").click  
end if  
'Above code will select the *third* checkbox on the webpage.

**Selecting the web check box using Description Programming -**

Set DescObject = Description.create()  
DescObject("micclass").value = "webcheckbox"  
  
set col = Browser("myb").page("myp").childobjects(DescObject)  
col(2).click

**Selecting the web check box using HTML DOM -**

set col = Browser("myb").page("myp").object.getElementsByTagName("input")

*'Please note that above statement will give you all the elements on the webpage having input tag.*

*You can check the innerhtml property of each element to see if the given input element's type is a checkbox or not.*

for i=0 to i<col.count

x = col(i).getAttribute("type")

if lcase(trim(x)) = "checkbox" then

col(i).checked = true

end if

next

**What are the challenges while automating the test cases in QTP?**

* Automation of third party controls like infragistic controls in .net based applications.
* Automation of the PDF file comparison.
* Working with the controls inside tables.
* Selecting the drop down value will not trigger the onchange event in webpage.
* Automation of the web applications involving frames/iframes/framesets.
* All edit boxes having same properties.
* Elements like checkboxes, buttons, radio buttons inside nested tables.

This is the list of most frequent challenges faced in automation projects. You can easily handle these challenges if you know how to access native properties of the objects.

QTP does provide the way to access native properties and methods using object property.

You can solve most of the issues in web application testing using html [DOM](http://qtp-interview-questions.blogspot.in/2013_06_01_archive.html)in QTP. Please try to avoid index to identify any object as this will make your script very weak.

**How to click on the links or buttons in webtable in QTP?**

Consider below web table. It has got 4 rows. Now what we have to do is – we have to click on the edit or delete link in front of the given employee.  
  
Employee Name Operation     
Sagar                Edit  / Delete     
Amol                Edit  / Delete     
ajit                    Edit  / Delete   
  
  
If we try to save links in OR, QTP uses the index property to identify the edit or delete buttons. But using index is not reliable because if new employee is added in the middle index of the existing links will also change.  
  
To handle such situations, we can use html dom. We can solve the problem using below approach.  
1. First get the row number in which given employee exists.  
2. Then find the links in that row using getElementsByTagName method.  
Here is the code to find the row number for the given employee.  
  
function getRow(byval Employee)  
Set tableObject = Browser(“MyB”).Page(“Myp”).WebTable("Mytable")  
For i=1 to tableObject.object.rows.count  
if tableObject.getCellData(i,1) = Employee Then  
print "row found " & i  
getRow = i  
exit function  
end if  
Next  
getRow = -1  
End function  
  
Above function will get the rownumber for the given employee. After this we can get the links in front of the given employee using below code.  
  
Set tableObject = Browser(“MyB”).Page(“Myp”).WebTable("Mytable")  
Set rowElement = tableObject.object.getElementsByTagName("tr")(rowNumber+1)  
set links = rowElement.getElementsByTagName("A")  
'Above statement will return all the links inside given row.  
if links(0).innerText = "Edit" then  
links(0).click()  
End If

**How to access excel as a database using adodb.connection in QTP?**

We can work with excel in 3 ways in QTP as mentioned below.

Using Excel.Application

Using Adodb.Connection

Using DataTable

To know how we can use Excel.Application to read or write the excel files.  
  
To know how to use datatable to load excel into it.  
  
In this article we will see how we can read or write excel files using Adodb.Connection object.  
Excel workbook is considered as a database. Each sheet in the excel workbook is considered as a table.  
First row in the sheet is considered as column header and all other rows are considered to be records.  
  
Here is the sample code to connect to excel database.  
  
Set excelConnection = createobject("Adodb.Connection")  
excelConnection.open "Data Source=c:\abc.xlsx;Provider=Microsoft.Jet.OLEDB.4.0"  
Set rs = excelConnection.execute "Select \* from [sheet1$]"  
For i=0 to rs.recordCount-1  
For j=0 to rs.fields.count-1  
print rs.fields(j).name & rs.fields(j).value  
Next  
Next

**How to trigger the web events like change, click, mouseup in QTP?**

Testing the web application with QTP involves the biggest challenge of events bound to the html elements with jquery by developers not getting triggered.

Example -

Problem (Issue)

For instance, when we select the value from the web combobox (dropdown), HTML dom of the webpage changes. It may change the visibility of the some of the controls like edit boxes or drop downs. It may do some ajax calls to the server by using javascript code.

So basically when we do some operation on the web elements , specific event handlers ( java script code or function) get called. But performing the same operation with QTP may not call these event handlers leading to inability to automate the test cases.

Solutions

To handle this issue we can trigger any event like change, click, double click, mouseup, mousedown etc by using FireEvent method. But if it does not work, you can use the vbscript code as depicted below.

'Create the event object

Set evt =  Browser("ABC").Page("XYZ").Object.createEvent("HTMLEvents")

'Set the event type like change, click, mouseup, mousedown

evt.initEvent "change", true, true

'Dispatch the event on desired object.

Browser("ABC").Page("XYZ").WebList("pqr").Object.dispatchEvent(evt)