## QTP Demo: A data driven Implementation

**Introduction**: QTP has some very practical usages which require very little time investment, encounter no tool complexities, but has huge benefits. The current topic demonstrates one such implementation.

Lets us say that you have a calculator application and you want to test all possible numeric combinations with all the available operators. I am just talking about the simple calculator (not scientific) Here are the test steps/cases:

Invoke the Application. Use all possible numeric combinations to perform all the possible calculations as mentioned below.

- 1. Multiplications (All possible combinations right from 1, 2, 3..... to 999999.....)
- 2. All possible Subtractions
- 3. All possible Additions
- 4. Divisions
- 5. Close the Application

⇒ Total number of manual Test cases would be in thousands and time taken would be in hours. Still **zero** accountability/proof whether the tester has actually executed the tests or spent the entire day playing video games (BTW, it does happen).

## In order to automate the test application, perform the following steps:

Prepare a matrix of all the possible numeric combinations with all the available operations (for example 12345679\*8=) in an excel sheet and save it in a predefined location.

- 1. Invoke QTP.
- 2. Click Record and invoke the calculator application. The following code will get generated *SystemUtil.Run "calc"*, *""*, *"C:\Documents and Settings\abhinav"*, *""*
- 3. Click view datatable.
- 4. In order to do the calculations of the test cases (multiple operator tests of various numerical data sets), type the following command: Window("Calculator").Type DataTable("DataTable123456798\_param\_dtGlobalSheet", dtGlobalSheet)

 Click File→Settings. The Test Settings dialog box as displayed in Figure 1 will get displayed.

Test Settings						
	Properties R	un Resources Parameters Environment Web Recovery				
	Name:	calc				
	Author:	abhinav				
	Created by:	QuickTest Professional 9.0				
	Last modified	1 by: QuickTest Professional 9.0				
	Location:					
	C:\Program I	Files\Mercury Interactive\QuickTest Professional\Tests\calc				
	Description					
Calculator: Data Driven Example						
	Associated add-ins:					
	ActiveX Visual Basic Web					
	Modify					
	Generate Script Click to generate an automation script for the test settings					
		OK Cancel Apply Help				

Figure 1: Test Settings

- 6. Click the **Run** tab. Following screen will be displayed. It has the radio buttons which need to be selected for the no of iterations you look forward to run.
- 7. Select run on all rows radio button.

Test Settings						
Properties Run Resources Parameters Environment Web Recovery	1					
Data Table iterations O Run on eliteration only Euron all rows D Run from row						
When error occurs during run session: pop up message box						
Object synchronization timeout: 20 seconds						
Save image of desktop when error occurs (if test is run by the Mercury Business Process Monitor)						
OK Cancel Apply Help						

Figure 2: Test Settings -- Run

8. Click the **Resources** tab. The following dialog box will be displayed. Under the data table option, browse to select the xls file created for test cases.

**Note**: There is a dependency on the naming convention of the excel file which needs to match the name of the action name specified in the code. QTP does prompt with an information dialog box as shown below.

QuickTest Professional					
The sheet names in the specified data table file should match the action names in yo The column headings in the data table sheets should match the parameter names in Otherwise, your test may fail.					
	OK				

 Close the Application/Calculator using Window("Calculator").Close. The QTP will have just 3 lines of Test code as displayed below:



Figure 3: Code Window

Run the test. **Observe/Record** the Results.

The entire test with all the specified data will get executed. The following result will get displayed after the successful completion of the tests.

🚰 calc [Res39] - Test Results		
Eile <u>V</u> iew <u>T</u> ools <u>H</u> elp		
। 🖆 🗇 🝸 🌏 🛠   ଲେ ପ୍ ସ୍   ← →   ?		
🖃 🖌 🏹 Test calc Summary	37	Passed
III Run-Time Data Table	<u> </u>	
🗄 🗸 🏹 calc Iteration 1 (Row 1)	38	Passed
🗄 🗸 🏹 calc Iteration 2 (Row 2)	39	Passed
✓ 🍅 calc Iteration 4 (Row 4)	<u>40</u>	Passed
E calc Iteration 5 (Row 5)	41	Passed
Calc Iteration 6 (Row 6)		
Calc Iteration 7 (Row 7)	42	Passed
E → ✓ 🎽 calc Iteration 9 (Row 9)	<u>43</u>	Passed
🗄 🖌 🖌 🏹 calc Iteration 10 (Row 10)	44	Passed
🗄 🗸 🏹 calc Iteration 11 (Row 11)	<u></u>	Tubbeu
🖅 🖌 🎲 calc Iteration 12 (Row 12)	<u>45</u>	Passed
🔃 🖌 🎲 calc Iteration 13 (Row 13)	46	Passed
🕀 🖌 🎲 calc Iteration 14 (Row 14)		
	4/	Passed
calc Iteration 16 (Row 16)	48	Passed
ealc Iteration 17 (Row 17)		
calc Iteration 18 (Row 18)	49	Passed
En Calc Iteration 20 (Row 20)	50	Passed
the state iteration 20 (Row 20)		Desered
E calc Iteration 22 (Row 22)	<u>51</u>	Passed
E v calc Iteration 23 (Row 23)	<u>52</u>	Passed
🗄 🖌 🍢 calc Iteration 24 (Row 24)	53	Passad
🕀 🖌 🚺 calc Iteration 25 (Row 25)	<u></u>	Fassed
🗄 🖌 👔 calc Iteration 26 (Row 26)	<u>54</u>	Passed
🗄 🖌 🏹 calc Iteration 27 (Row 27)	55	Passed
🕀 🖌 👔 calc Iteration 28 (Row 28)	<u> </u>	1 03500
🗈 🖌 🎽 calc Iteration 29 (Row 29)	<u>56</u>	Passed
🛛 🕀 🖌 🖌 🖌 🖌 🖌 🖌 🖌		
For Help, press F1		Ready

Figure 4: Test Results

That's the fun of doing Automation !!!