

# QTP - DESIGNING FRAMEWORK

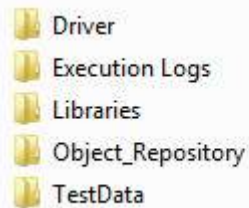
[http://www.tutorialspoint.com/qtp/qtp\\_designing\\_framework.htm](http://www.tutorialspoint.com/qtp/qtp_designing_framework.htm)

Copyright © tutorialspoint.com

## Designing a Framework

The sample application under test is "**Calculator**", a default application that is available as part of Windows. Let us now create different components of a framework. Here, we will develop a hybrid framework and use Object Repository as it is fairly a simple application. However, this framework can be scaled to support a complex application as well.

The Folder Structure of the Framework is as follows. The Explanation of the folder structure is explained below:



- Master Driver Script - The Script that drives the entire execution. It performs prerequisite and initial settings that are required for the execution.
- Library Files - The Associated Functions that forms the Function Library
- Data Table - The Test Data that is required for the Execution.
- Object Repository - The Objects and its properties that enable QTP to recognize the objects seamlessly.
- Execution Logs - The Folder contains the execution log file with user functions and function execution history.

## Master Driver Script

```
'=====
' MASTER DRIVER SCRIPT NAME      : Calculator
' DESCRIPTION                    : Drivers Script to Execute Automated Test for the
Calculator
' AUTHOR                        : Tutorials Point
' DATE CREATED                  : 30-Dec-2013
' OBJECT REPOSITORY ASSOCIATED  : Calc.tsr
' LIBRARY FILES ASSOCIATED      : calculator.qfl, Common.qfl
' MODIFICATION LOG
' -----
' First Version      Tutorials point
'=====
Option Explicit

Public ExecDrive

' Get the Root folder of the Test so that we can make use of relative paths.
Dim x : x=Instr(Environment.Value("TestDir"),"Driver")-2
ExecDrive = mid(Environment.Value("TestDir"),1,x)

' Get the path of Libraries using relative to the current Drive
Dim LibPath : LibPath = ExecDrive+"\Libraries\"

' Dynamically Load the Function Libraries
LoadFunctionLibrary LibPath + "Calculator.qfl", LibPath + "common_utils.qfl"

' Capturing the Start Time
' clscommon is the class object created in common.qfl library file
```

```

clscommon.StartTime = Time()

' Launching the Application
SystemUtil.Run "C:\Windows\System32\Calc.exe" : wait (2)

' Initialize the Data Table Path
Dim FileName : FileName = ExecDrive+"\TestData\Calculator.xls"
Dim SheetSource : SheetSource = "Calc_test"
Dim SheetDest : SheetDest = "Global"

' Import the DataTable into the QTP Script
DataTable.ImportSheet FileName , SheetSource , SheetDest

' Object Repository Path
Dim RepPath : RepPath = ExecDrive+"\Object_Repository\Calc.tsr"
RepositoriesCollection.RemoveAll()
RepositoriesCollection.Add(RepPath)

' To Keep a Count on iteration
Dim InttestIteration
Dim InttestRows : InttestRows = datatable.GetRowCount

' Fetching Date-TimeStamp which will be unique for Naming the Execution Log File
clscommon.StrDateFormatted = day(date()) & "_" & MonthName(Month(date()), true) &
    "_" & YEAR(date()) & "_" & hour(now) & "_" & minute(now)

' Name the LogFile
clscommon.StrLogFile = ExecDrive & "\Execution Logs\" &
clscommon.StrDateFormatted & ".txt"

' Create the Execution LogFile which captures the result
clscommon.Fn_FileCreate(clscommon.StrLogFile)

' Initialize the Parameters and all the relevant Test Details
Call Fn_InitializeLogFile()

' Kill all the previous calculator process
Call fn_Kill_Process("calc.exe")

For InttestIteration=1 to InttestRows
    datatable.SetCurrentRow InttestIteration
    Dim StrExecute : StrExecute = Ucase(Trim(datatable.Value("Run", "Global")))
    If StrExecute = "Y" Then
        clscommon.Number1 = Trim(datatable.Value("Number_1", "Global"))
        clscommon.Number2 = Trim(datatable.Value("Number_2", "Global"))
        clscommon.Number3 = Trim(datatable.Value("Number_3", "Global"))

        clscommon.Number4 = Trim(datatable.Value("Number_4", "Global"))
        clscommon.Number5 = Trim(datatable.Value("Number_5", "Global"))
        clscommon.Number6 = Trim(datatable.Value("Number_6", "Global"))

        clscommon.Test_Case_ID = Trim(datatable.Value("Test_Case_ID", "Global"))'
:   clscommon.LogWrite "The Test Case Data is Located at :: " & tcDataPath
        clscommon.tcScenario = Trim(datatable.Value("Scenario", "Global"))'
:   clscommon.LogWrite "The Test Case Data is Located at :: " & tcDataPath
        Dim Expected_Val : Expected_Val =
Trim(datatable.Value("Expected_Val", "Global"))'
        clscommon.LogWrite "The Test Case Data is Located at :: " & tcDataPath

        Select case clscommon.tcScenario
            Case "Add"
                clscommon.LogWrite "=== Inside the Test Set :: " & clscommon.tcScenario & "
                Call fnCalculate("+", Expected_Val)

            Case "Subtract"
                clscommon.LogWrite "=== Inside the Test Set :: " & clscommon.tcScenario & "
                Call fnCalculate("-", Expected_Val)

```

```

        Case "Multiply"
            clscommon.LogWrite "=== Inside the Test Set :: " & clscommon.tcScenario & "
===
            Call fnCalculate("*",Expected_Val)

        Case "Divide"
            clscommon.LogWrite "=== Inside the Test Set :: " & clscommon.tcScenario & "
===
            Call fnCalculate("/",Expected_Val)

        Case "Sqrt"
            clscommon.LogWrite "=== Inside the Test Set :: " & clscommon.tcScenario & "
===
            Call fnCalculate("sqrt",Expected_Val)

    End Select
End If
Next

' Calling the End Test to Add the result Footer in exec log file.
Call fn_End_test()

' ===== End of Master Driver Script =====

```

## Library Files

The Calculator Functions are written in a separate function file saved with the extension .qfl or .vbs. These functions are reusable across actions.

```

' Calculator. Qfl File :: Associated Function Library for Calculator Master Driver

'= = = = =
' FUNCTION NAME      : Fn_InitializeLogFile
' DESCRIPTION        : Function to Write the Initial Values in the Log File
' INPUT PARAMETERS   : varExecDrive,StrDB,StrUIId,Strpwd,StrNewDB
' OUTPUT PARAMETERS  : NIL
' RETURN VALUE       : Pass or Fail message
' DATE CREATED       : 30-Dec-2013
'= = = = =

Public Function Fn_InitializeLogFile()
    clscommon.LogWrite "*****"
    clscommon.LogWrite "Calc Automation Started"
End Function

'= = = = =
' FUNCTION NAME      : fnCalculate
' DESCRIPTION        : Function to perform Arithmetic Calculations
' INPUT PARAMETERS   : operator,Expected_Val
' OUTPUT PARAMETERS  : NIL
' RETURN VALUE       : Pass or Fail message
' DATE CREATED       : 30-Dec-2013
'= = = = =

Function fnCalculate(operator,Expected_Val)
    clscommon.LogWrite "Executing the Function 'fnCalculate' "

    Window("Calculator").Activate

    If Trim(clscommon.Number1) <> "" Then
Window("Calculator").WinButton(clscommon.Number1).Click
    If Trim(clscommon.Number2) <> "" Then
Window("Calculator").WinButton(clscommon.Number2).Click
    If Trim(clscommon.Number3) <> "" Then
Window("Calculator").WinButton(clscommon.Number3).Click

    Window("Calculator").WinButton(operator).Click

    If Trim(clscommon.Number4) <> "" Then
Window("Calculator").WinButton(clscommon.Number4).Click

```

```

If Trim(clscommon.Number5) <> "" Then
Window("Calculator").WinButton(clscommon.Number5).Click
If Trim(clscommon.Number6) <> "" Then
Window("Calculator").WinButton(clscommon.Number6).Click

Window("Calculator").WinButton("=").Click
Dim ActualVal : ActualVal =
Window("Calculator").WinEdit("Edit").GetROProperty("text")
clscommon.LogWrite "The Actual Value after the Math Operation is "& ActualVal

If Trim(ActualVal) = Trim(Expected_Val) Then
clscommon.WriteResult "Pass", clscommon.Test_Case_ID , clscommon.tcScenario , "
Expected Value matches with Actual Value :: " & ActualVal

Else
clscommon.WriteResult "Fail", clscommon.Test_Case_ID , clscommon.tcScenario , "
Expected Value - " & Expected_Val & " Does NOT matches with Actual Value :: " &
ActualVal
End If

Window("Calculator").WinButton("C").Click

If Err.Number <> 0 Then
clscommon.LogWrite "Execution Error : The Error Number is :: " & Err.Number & "
The Error Description is " & Err.Description
Err.Clear
End If

clscommon.LogWrite "Exiting the Function 'fnCalculate' "
End Function

'= = = = =
' FUNCTION NAME : fn_Kill_Process
' DESCRIPTION : Function to Kill the process by name
' INPUT PARAMETERS : Application name to be killed
' OUTPUT PARAMETERS : NIL
' RETURN VALUE : NIL
' DATE CREATED : 30-Dec-2013
'= = = = =
Function fn_Kill_Process(process)
Dim strComputer , strProcessToKill , objWMIService , colProcessesstrComputer = "."
strProcessToKill = process

Set objWMIService = GetObject("winmgmts:" _& "{impersonationLevel=impersonate}!\" _&
strComputer & "\root\cimv2")

Set colProcess = objWMIService.ExecQuery _& ("Select * from Win32_Process Where Name =
'" & strProcessToKill & "'")

count = 0
For Each objProcess in colProcess
objProcess.Terminate()
count = count + 1
Next
End Function

'= = = = =
' FUNCTION NAME : fn_End_test
' DESCRIPTION : Function to finish the test Execution process
' INPUT PARAMETERS : Application name to be killed
' OUTPUT PARAMETERS : NIL
' RETURN VALUE : NIL
' DATE CREATED : 20/Dec/2013
'= = = = =
Function fn_End_test()
clscommon.LogWrite "Status Message - Executing the Function 'fn_End_test' "

Window("Calculator").Close
On Error Resume Next

```







```

8/Jan/2014 5:09:16 PM: *****
8/Jan/2014 5:09:16 PM: Calc Automation Started
8/Jan/2014 5:09:16 PM: === Inside the Test Set :: Add ===
8/Jan/2014 5:09:16 PM: Executing the Function 'fnCalculate'
8/Jan/2014 5:09:17 PM: The Actual Value after the Math Operation is 949.
8/Jan/2014 5:09:17 PM: * * * * * Test Case Exec Details * * * * *
8/Jan/2014 5:09:17 PM: Test staus :: Pass
8/Jan/2014 5:09:17 PM: Tese ID :: TC_001
8/Jan/2014 5:09:17 PM: Test Description :: Add
8/Jan/2014 5:09:17 PM: Test Result Details :: Expected Value matches with Actual Value
:: 949.
8/Jan/2014 5:09:17 PM: * * * * *
* * * * *
8/Jan/2014 5:09:17 PM: Exiting the Function 'fnCalculate'
8/Jan/2014 5:09:17 PM: === Inside the Test Set :: Subtract ===
8/Jan/2014 5:09:17 PM: Executing the Function 'fnCalculate'
8/Jan/2014 5:09:17 PM: The Actual Value after the Math Operation is 415.
8/Jan/2014 5:09:17 PM: * * * * * Test Case Exec Details * * * * *
8/Jan/2014 5:09:17 PM: Test staus :: Pass
8/Jan/2014 5:09:17 PM: Tese ID :: TC_002
8/Jan/2014 5:09:17 PM: Test Description :: Subtract
8/Jan/2014 5:09:17 PM: Test Result Details :: Expected Value matches with Actual Value
:: 415.
8/Jan/2014 5:09:17 PM: * * * * *
* * * * *

```

```
8/Jan/2014 5:09:17 PM: Exiting the Function 'fnCalculate'
8/Jan/2014 5:09:17 PM: === Inside the Test Set :: Multiply ===
8/Jan/2014 5:09:17 PM: Executing the Function 'fnCalculate'
8/Jan/2014 5:09:18 PM: The Actual Value after the Math Operation is 278883.
8/Jan/2014 5:09:18 PM: * * * * * Test Case Exec Details * * * * *
8/Jan/2014 5:09:18 PM: Test staus :: Pass
8/Jan/2014 5:09:18 PM: Tese ID :: TC_003
8/Jan/2014 5:09:18 PM: Test Description :: Multiply
8/Jan/2014 5:09:18 PM: Test Result Details :: Expected Value matches with Actual Value
:: 278883.
8/Jan/2014 5:09:18 PM: * * * * *
* * * * *

8/Jan/2014 5:09:18 PM: Exiting the Function 'fnCalculate'
8/Jan/2014 5:09:18 PM: === Inside the Test Set :: Divide ===
8/Jan/2014 5:09:18 PM: Executing the Function 'fnCalculate'
8/Jan/2014 5:09:19 PM: The Actual Value after the Math Operation is 3.
8/Jan/2014 5:09:19 PM: * * * * * Test Case Exec Details * * * * *
8/Jan/2014 5:09:19 PM: Test staus :: Pass
8/Jan/2014 5:09:19 PM: Tese ID :: TC_004
8/Jan/2014 5:09:19 PM: Test Description :: Divide
8/Jan/2014 5:09:19 PM: Test Result Details :: Expected Value matches with Actual Value
:: 3.
8/Jan/2014 5:09:19 PM: * * * * *
* * * * *

8/Jan/2014 5:09:19 PM: Exiting the Function 'fnCalculate'
8/Jan/2014 5:09:19 PM: === Inside the Test Set :: Sqrt ===
8/Jan/2014 5:09:19 PM: Executing the Function 'fnCalculate'
8/Jan/2014 5:09:20 PM: The Actual Value after the Math Operation is 10.
8/Jan/2014 5:09:20 PM: * * * * * Test Case Exec Details * * * * *
8/Jan/2014 5:09:20 PM: Test staus :: Pass
8/Jan/2014 5:09:20 PM: Tese ID :: TC_005
8/Jan/2014 5:09:20 PM: Test Description :: Sqrt
8/Jan/2014 5:09:20 PM: Test Result Details :: Expected Value matches with Actual Value
:: 10.
8/Jan/2014 5:09:20 PM: * * * * *
* * * * *

8/Jan/2014 5:09:20 PM: Exiting the Function 'fnCalculate'
8/Jan/2014 5:09:20 PM: Status Message - Executing the Function 'fn_Finish_test'
8/Jan/2014 5:09:20 PM: ## # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#
8/Jan/2014 5:09:20 PM: ## The Execution Start Time :: 5:09:14 PM
8/Jan/2014 5:09:20 PM: ## The Execution End Time :: 5:09:20 PM
8/Jan/2014 5:09:20 PM: ## The Time Elapsed :: 0 Minutes
8/Jan/2014 5:09:20 PM: ## The OS :: Microsoft Windows Vista Server
8/Jan/2014 5:09:20 PM: ## The Total No of Test Cases Executed :: 25
8/Jan/2014 5:09:20 PM: ## The No. of Test Case Passed :: 25
8/Jan/2014 5:09:20 PM: ## The No. of Test Case Failed ::
8/Jan/2014 5:09:20 PM: ## # # # # # # # # # # # # # # # # # # # # # # # # # # #
#
```