

# LOGIGEAR

---

MOBILE TEST AUTOMATION WITH  
TESTARCHITECT™

## TABLE OF CONTENTS

1. Background.....	3
2. 5 Steps of Automated Test Execution with TestArchitect.....	4
3. Turnkey Solutions.....	6
4. Conclusion.....	6

## BACKGROUND

TestArchitect, (TA) LogiGear's premiere automation tool, has three major components:

### TestArchitect Client

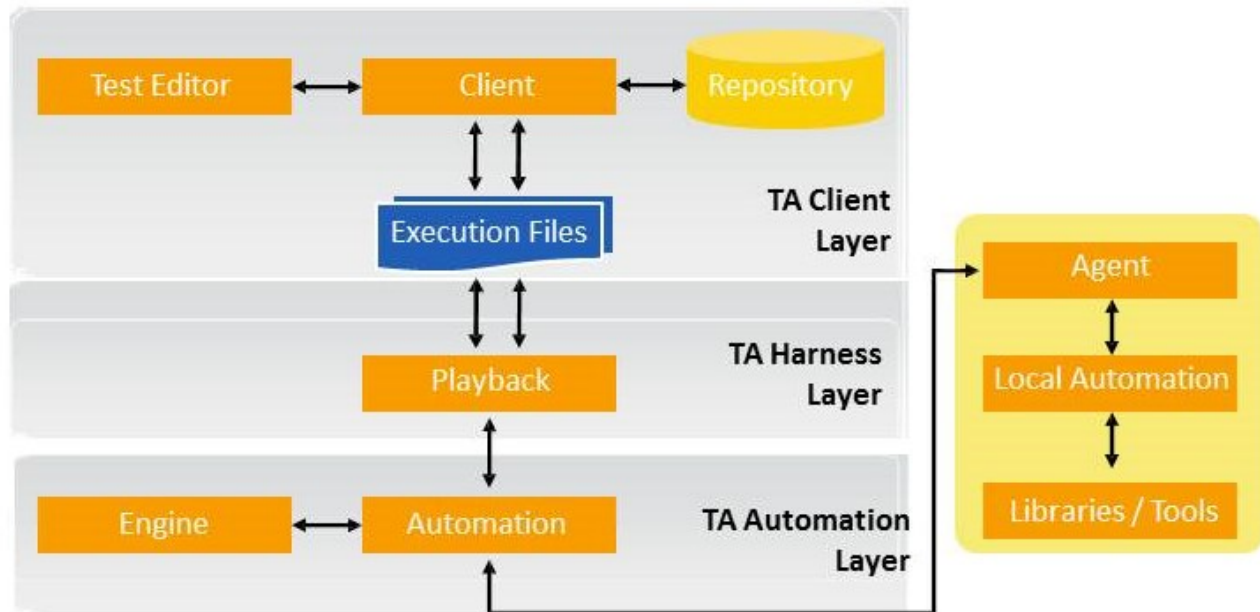
The client is the part of the application that interacts with the user directly. This includes the TestArchitect Explorer and other GUI components that you can see from the application. TA Client is also responsible for compiling a test into text files during a text execution.

### TestArchitect Harness

The Harness is the main entry point of a test execution. It reads the compiled text files line by line, and then feeds them to the automation engine. Custom actions can be implemented in the harness, and in this manner become available to the tests.

### Test Architect Automation

The Automation Libraries, as the name suggests, is the brains of the operation. It translates high-level actions into low-level functions, takes care of the control flow of the test execution, and interacts with the SUT and the Agent.



## 5 STEPS OF AUTOMATED TEST EXECUTION WITH TESTARCHITECT

1. TA Client compiles the test module being run into text files. It also creates a setting file called 'run.dat' that contains the settings of the test, and where the harness can find the text files.
2. After compiling, TA Clients will call the harness. The harness will use the 'run.dat' file to know which file/test to be run, and reads the file line by line.
3. If a running action matches a custom action defined in the harness, it will execute the action itself. Otherwise, it will pass it to the Automation engine.
4. The Automation engineer will convert actions into real programming functions, and then execute them.
5. The Automation engineer will forward the action to an active Agent.

## DELIVERED WITH CUSTOMIZED, TURNKEY AUTOMATION SOLUTIONS

TestArchitect can work with turnkey test automation based on your existing test cases, or TestArchitect can help you.

```
' register the mobile specific actions
Public Sub SetMobileActions()
  ' specify category
  SetActionActionCategory "mobile actions"
  ' specify action
  SetDirect "hello mobile"
End Sub
```

Registration

```
' divert to the action function
Public Sub DivertMobileAction(Action As String)
  Select Case Action
    Case "hello mobile": action_HelloMobile
    Case Else
      ABT.Problem ("Action " & Action & " not recognized")
  End Select
End Sub
```

Redirection

```
' action implementation
Public Sub action_HelloMobile()
  ' send the action to mobile 'main'
  SendMobile main, "hello"
End Sub
```

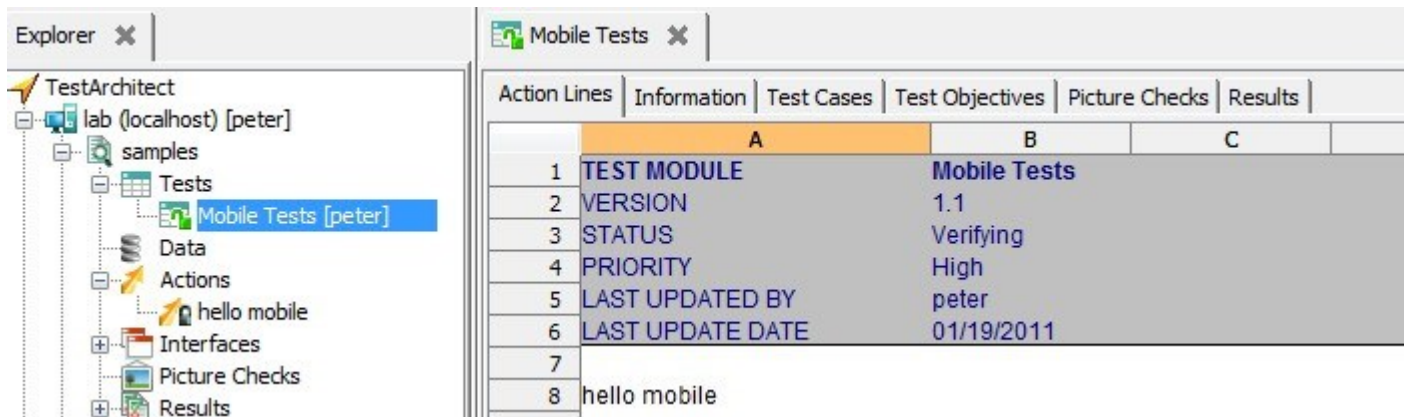
Implementation

```
' main entry point for the execution
Sub Main()
  ' load the mobile actions
  SetMobileActions
```

Register the 'mobile actions'  
code module

```
' this module is called by the Engine
Sub Divert(Module As String, Action As String)
  Select Case Module
    Case "mobile actions": DivertMobileAction Action
    Case ExecuteNow: ABTLIB.ExecuteNow
    Case "abt automation": ABT.Divert Action
  End Select
End Sub
```

Redirect the action to the  
'mobile actions' module



The screenshot shows the TestArchitect interface. On the left is the Explorer pane with a tree view containing 'lab (localhost) [peter]', 'samples', 'Tests', 'Mobile Tests [peter]', 'Data', 'Actions', 'hello mobile', 'Interfaces', 'Picture Checks', and 'Results'. The 'Mobile Tests [peter]' item is selected. On the right is the 'Mobile Tests' window with a table of information.

	A	B	C
1	TEST MODULE	Mobile Tests	
2	VERSION	1.1	
3	STATUS	Verifying	
4	PRIORITY	High	
5	LAST UPDATED BY	peter	
6	LAST UPDATE DATE	01/19/2011	
7			
8	hello mobile		

After compiling the harness, a test module can be created to use the new action.

The new action has now become part of the “test language”, and can be used in any number of tests as often as needed. And any changes in the system under test can be incorporated in the action script, generally without a need to revisit tests.

## LOGIGEAR’S SOLUTIONS

LogiGear is a leading provider of software testing services, test automation and application development and maintenance. We help organizations deliver better products while saving time and money. Since 1994, we have completed testing projects with hundreds of companies from early stage start-ups to Fortune 100, across a wide range of industries and technologies. For more information visit [Logigear.com](http://Logigear.com) or contact us at (650) 574-1400, or email us at [sales@logigear.com](mailto:sales@logigear.com).