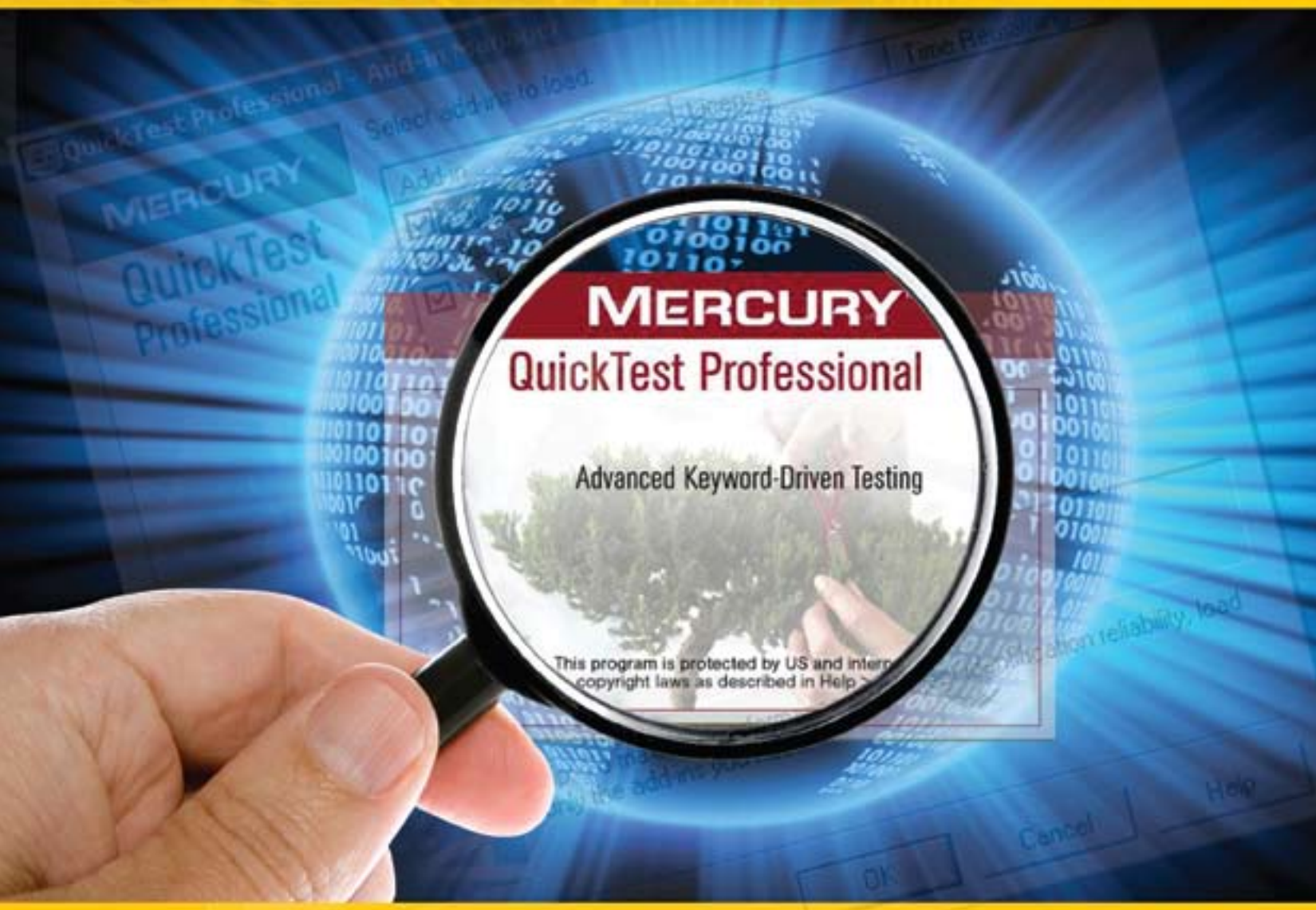
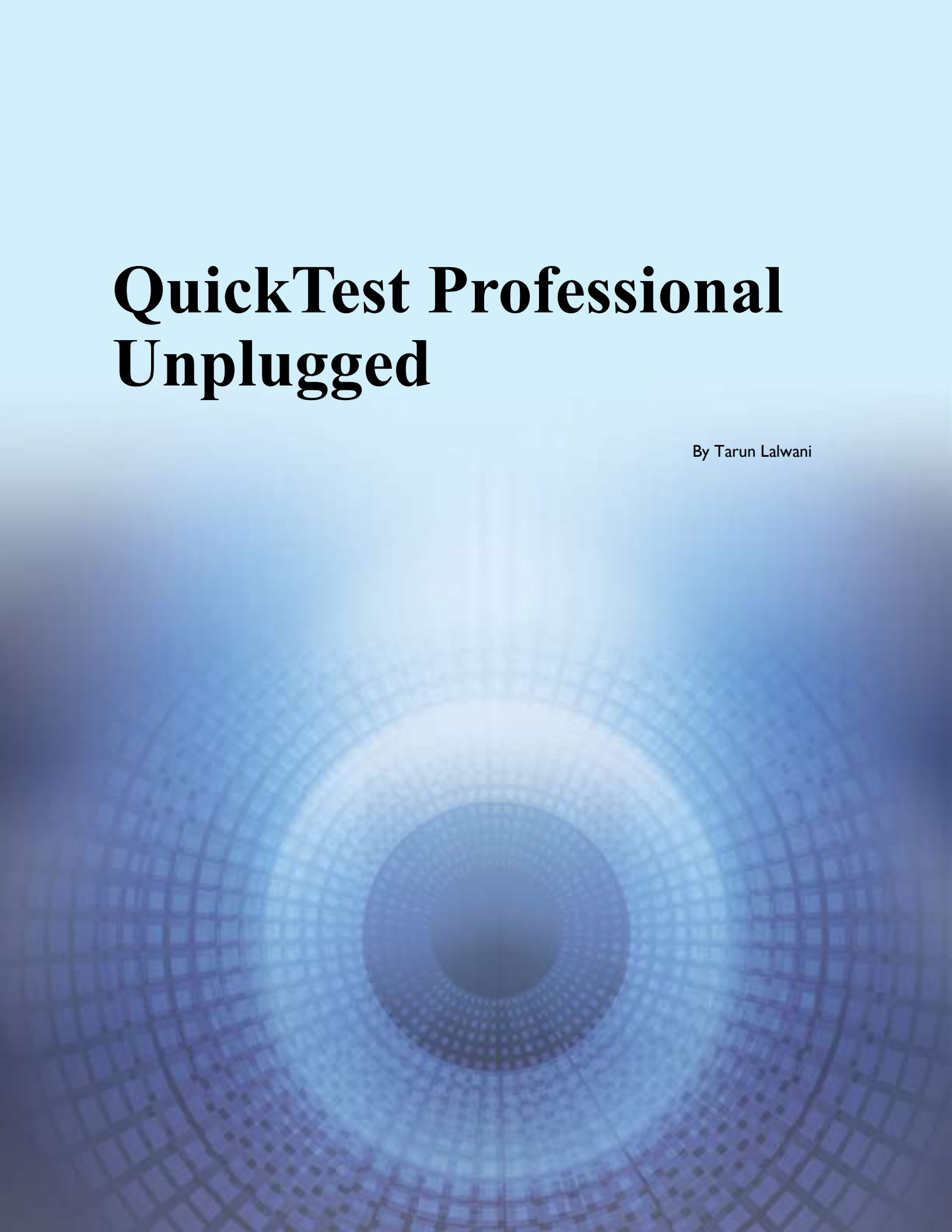
QuickTest Professional Unplugged



Tarun Lalwani

QuickTest Professional Unplugged

By Tarun Lalwani



QuickTest Professional Unplugged

By *Tarun Lalwani*

Editor: *Siddharth N Kanoujia*

Technical Editor: *Terry Howarth*

Reviewer: *Mark Smith and Terry Howarth*

Printing History:

May 2009: First edition

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise without either the prior written permission of the Author, or authorization through payment of the appropriate per-copy fee to the Author. For permission please contact author at *http://knowledgeinbox.com/contact-me*.

Source codes discussed in this book can be copied, modified or distributed without permission from author or the publisher by including the below mentioned reference

'Source code taken from "QuickTest Professional Unplugged" By Tarun Lalwani

'Website: *http://KnowledgeInbox.com/books/quicktest-professional-unplugged/*

This document also contains registered trademarks, trademarks and service marks that are owned by their respective companies or organizations. The Publisher and the author disclaim any responsibility for specifying which marks are owned by which companies or organizations.

Copyright (2008) Mercury Interactive (Israel) Ltd. Reproduced with permission

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

Contents

Preface

I started with Web testing Automation in mid 2004 by creating a project in Visual Basic 6 using the Internet explorer COM Automation. The project was a great success but with every change to the application the Automation Code had to be updated and recompiled. Trying to find a solution to this maintenance issue I stumbled upon QuickTest Professional 8.0 (QTP).

QuickTest Professional is a Test Automation tool and uses VBScript as its scripting language. QTP is a record and playback tool which can record events we perform on an application and replay them back. QTP is an object based tool which recognizes each element of the application as an object and provides various methods to work on them. All this makes look QTP an easy to use test tool. The myth about Record & Playback is that it makes people think that they do not need development skills for QTP, but to create effective Automation Frameworks one needs to view QTP as a development tool and not as a testing tool. This book presents QTP as a development tool rather than a mere test tool.

One of my problems while evaluating the tool led to me to join www.SQAForum.com, without knowing that I will specialize in the use of this tool in future. After sometime I launched KnowledgeInbox.com for sharing my articles on QTP with the larger group. Dealing with day to day automation problems faced by people on the QTP forums, I tried solving those problems for them and learnt a few new things on my own. Observing the patterns of queries being asked on the QTP forums, I thought what the QTP community was missing is a book which can guide the amateur automation engineers in becoming a professional in the use of this tool. I took up this responsibility and started writing this book in May 2005. I spent an year on the research of the undocumented QTP features and solving the unsolved queries of QTP.

Being a first time author, I had a very hard time getting this project completed. It was an additional responsibility, over and above my office work, QTP forum support, writing articles on KnowledgeInbox, creating tools for the community. It required a lot of motivation to keep myself on the project. But knowing what difference this book can make to the QTP community always kept me motivated.

I have organized the chapter in such a way that can make learning QTP an easier task. Each chapter is based on a QTP feature. The book is divided in two sections, Basics and Advanced. Chapter 1 to 18 cover features related to QTP while the chapters in the Advanced section cover integration/interaction of QTP with various external tools like Outlook, Word, Excel and Quality Center. The book discusses a lot of issues that are commonly faced while using various features of QTP and their resolution. This book discusses almost all the topics of QTP which one would require to create complex frameworks.

Who This Book Is For

This book is for Test engineers, Test Analysts, Test Consultants, Test Managers and anyone who is interested in learning advanced techniques of problem solving in QTP. This book is also for beginners who have just started with QTP and want to be experts in its use. The book assumes that one has the basic knowledge of QTP and VBScript, if not than it is advised that one should go through the basic help first. As the main focus of this book is to view the tool from a developer's eye, the book does not teach how to record and replay script in QTP. Also the book does not discuss about the Keyword view of QTP, which is for non-technical people who don't want to code in QTP.

Feedback and Queries

For any feedback or queries you contact the author at *http://KnowledgeInbox.com/contact-me* or post a query on the KnowledgeInbox forums – *http://Knowledgeinbox.com/forums/*

Acknowledgements

The following individuals deserve a special mention for reviewing my work in progress, and providing me with many useful comments and suggestions:

Mark smith, Terry Horwath and Siddharth N Kanoujia

Mark works as a freelance contractor and can be contacted on *Quicktest@gmail.com*

Terry has not only worked as a reviewer but also as a Technical editor for this book. Terry has worked with automated testing tools since the early 90's when he started designing solutions with Segue's QA Partner 1.0. He transitioned to Mercury Interactive tools in 1999 and has worked exclusively with QuickTest Professional since 2005. He can be contacted on *thorwath@lakefolsom.com*.

Siddharth has worked as a editor for this book and has done a great job making sure that contents of this book are lucid and unambiguous for the beginners. He is an Assistant Professor in the Department of English at Hindu College, University of Delhi. He can be contacted on *kandidsid@gmail.com*

I would especially like to thank my family and friends who have always motivated me while I was working on this book.

Quotes From Reviewers

"I find this to be a very pragmatic, hand's on book for those who want to extend their QTP skills beyond basic expert view programming. This book is written by a QTP master for those who wish to eventually become masters themselves." **– Terry**

"Tarun Lalwani has singlehandedly helped thousands of people to expand their knowledge of QuickTest Professional. Here is a book the automated testing community has been crying-out for. This book will help QTP practitioners, from beginner to expert. I have used QTP from V6.0 and during the review I learnt something from every chapter." **– Mark**

"After long brainstorming sessions with Tarun over almost each and every sentence, I realized that I am truly in the presence of a genius. Tarun has with his dedication and perseverance made possible a book which will go a long way in helping people understand the ins and outs of QTP." **– Siddharth**

Chapter 4

DataTables

A DataTable provides a way to create data driven test cases. It is similar to MS Excel spreadsheets and can be used to run an Action multiple times. Each test case has one global data sheet which is accessible to all actions inside that test case and each action has its own private data table also known as local data table. The name local data table is somewhat misleading because it is in fact possible to access any action's local data table from any other action, but the way of accessing the data becomes a bit different.

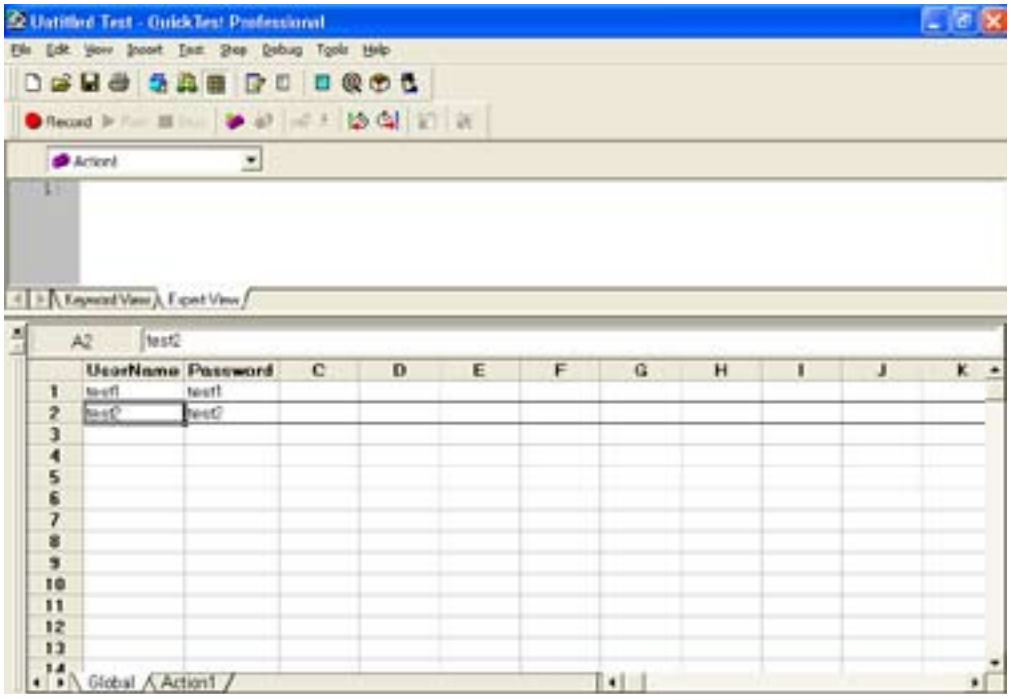


Figure 4-1. DataTable

Figure 4-1 shows a sample DataTable with 2 parameters, Username and Password.

We can use most of the formulas that work inside a typical Excel spreadsheet. But there are some differences between a DataTable and an Excel spreadsheet. In fact a DataTable is wrapped around an Excel spreadsheet—which provides access functionality to the values but does not expose the Excel spreadsheet object model.

```
'gives the value of Parameter1 stored in
'the Global data table.
DataTable("Parameter1",dtGlobalSheet)

'gives the value of Parameter1 stored in
'the current's action local data table.
DataTable("Parameter1",dtLocalSheet)
```

The same DataTable cannot have duplicate parameter names but we can use the same name parameters in different sheets (Global DataTable and Local DataTable). Each DataTable has only 1 row enabled even when it is blank and the other rows get enabled when data is entered into a new row. A DataTable is stored as "Default.xls" file in the test folder. Figure 4-2 shows how the stored file looks like

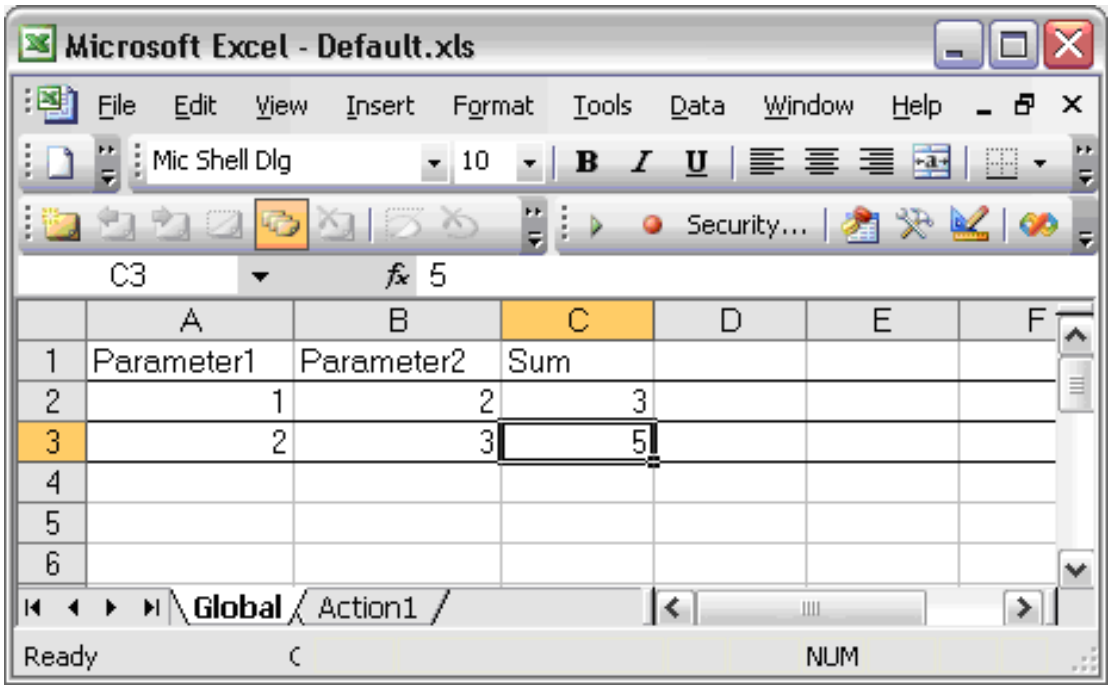


Figure 4-2. Default.xls Data table

When viewed in Excel, the first row of the sheet contains the parameter names, while QTP displays the parameter as the column titles. Therefore, when viewed using Excel, the 2nd row starts the 1st row of data in the DataTable. The DataTable shown above has only 2 data rows enabled. Note that QTP makes a data row enabled by marking the borders of the row in the actual spreadsheet. A row with no data but with marked borders is still considered as an enabled row by QTP. To delete an enabled row we must select the row and delete it from the context menu which appears on right clicking the row.

Design and run-time data table

Design time data table

As the name suggest the data table during the script design time is known as design time data table. Any changes to this are saved when the script is saved.

Run-time data table

The run-time data table contains a copy of the design time data table when a script is executed. It may contain values that are changed during script execution and are presented in the test result summary. The changes made to the data table during run-time are not saved to design time data table. Figure 4-3 shows a run-time data table from the test results summary

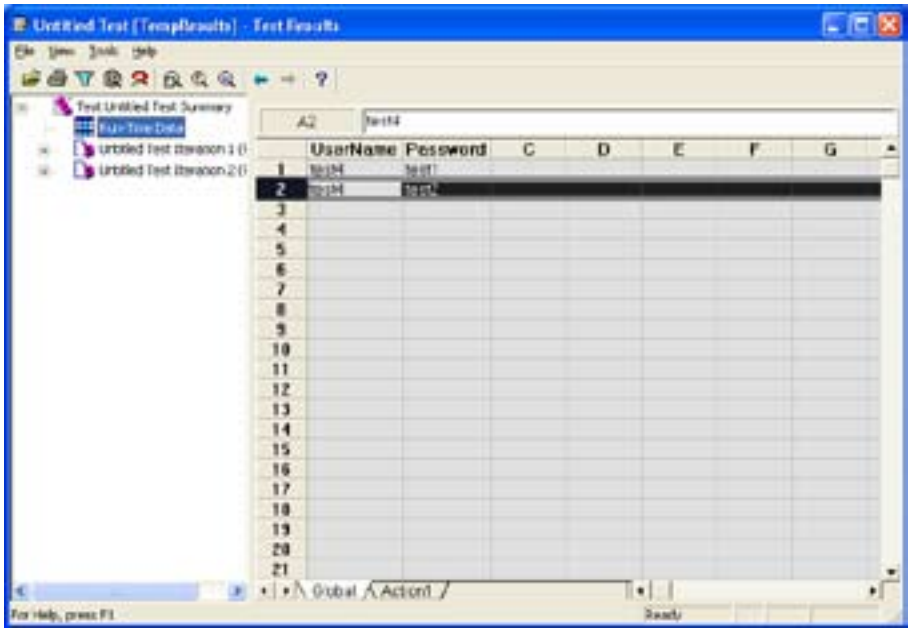


Figure 4-3. Run-time data table

When to use the global or a local data table

It is important to understand in what situations the global or a local data table should be used. Consider the following two scenarios

Scenario 1 - Log into the application, book 1 ticket, log out. Repeat the scenario for many users

Scenario 2 - Log into the application, book 3 tickets, and log out

Scenario 1

The Global data table is better suited for this scenario where we have the user name, password and tickets details as the parameters and we execute the scenario using a single action (which does everything) or multiple actions (Login, booking and logout).

Scenario 2

A Local data table is better suited for this scenario. Here a good approach would be to split the test into three actions: login, booking and logout. Login and logout can use the username and password parameters from the global data table and booking can use ticket detail parameters from its local data table and the action will be executed for all rows in its local data table.

Setting data table iterations

To run a test case for some number of iterations we need to set the iterations of global data table in the Test Settings dialog, which is invoked using *Test→Settings…→Run (Tab)* Figure 4-4 shows the iteration settings for the global table. These settings are specific to script.

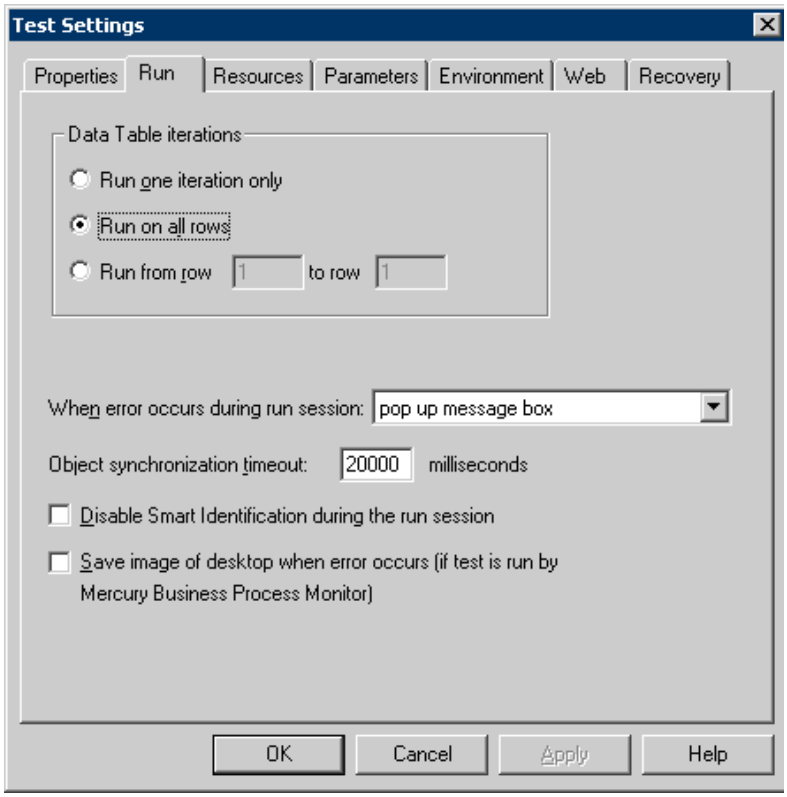


Figure 4-4. Global data iterations

NOTE: We can use an external spreadsheet as a Data table by specifying the location of the file in the *Resource (Tab)* as shown in the Figure 4-4

We can set the iteration settings for an Action call by going into the keyword view and then right clicking on the Action and selecting *Action Call Properties…*as shown in the Figure 4-5

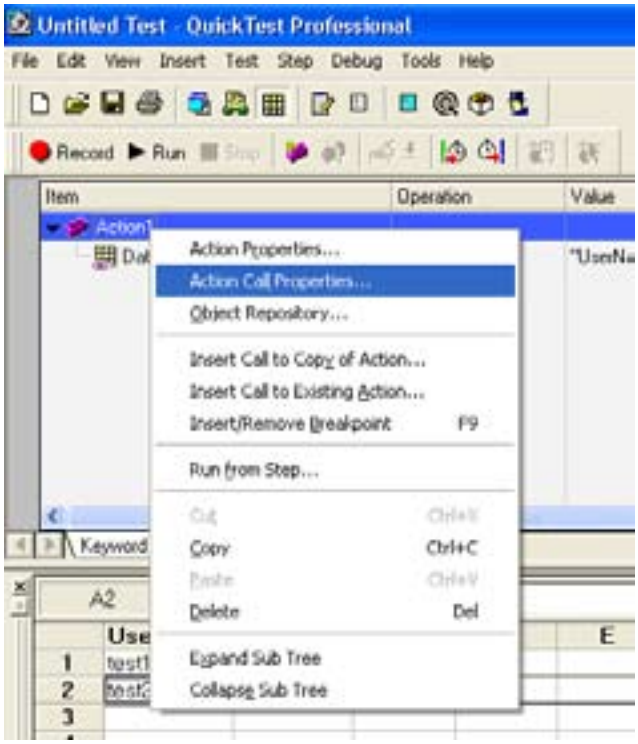


Figure 4-5. Action Call Properties - Keyword View

The Action call properties dialog can be used to set the iterations as shown in the Figure 4-6

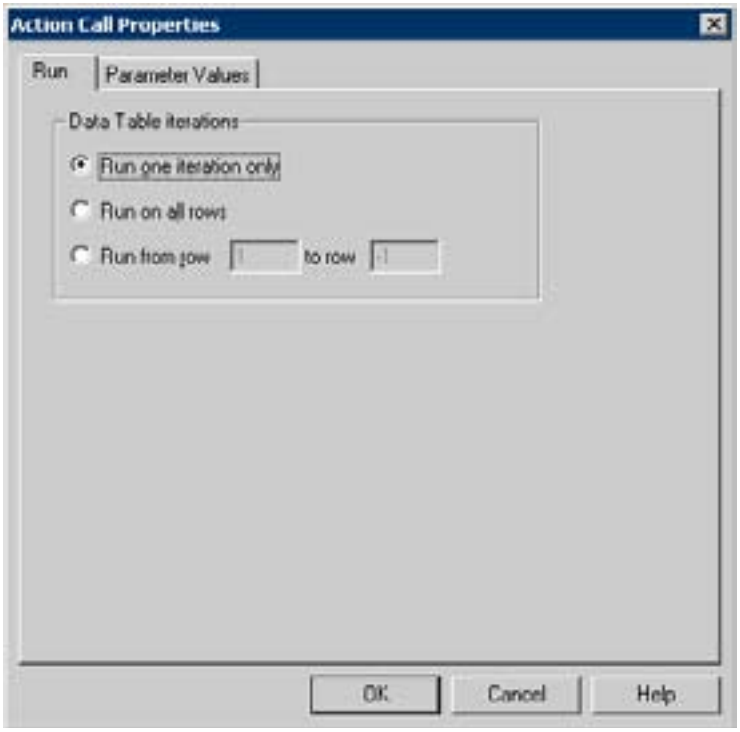


Figure 4-6. Action Call Properties dialog

Data table object model

QTP provides an object model to access various properties and methods in a data table:

There are three types of objects

- DataTable – Represents all the global and local data tables in the test
- DTSheet – Represents a single sheet in the test
- DTParameter – Represents a single column in a sheet.

Each object has certain functions available and certain properties associated with it. These are explained in detail in the QTP user manual.

Data table formatting

When data is entered into the data table it automatically formats the value using the best possible matching format. For example, if "12345678901" is entered into a cell then it would be auto formatted to "1.23456789E+010". In situations where the formats are important the data should be entered with care. If data entered in the cell start with a single quote (') then it is always treated as a text and no format conversion is performed.

We can also define a specific format by right clicking the cell or an entire column and then picking a specific format from the popup context menu.

Problem 4-1. How to access a parameter from the global data sheet

There are a variety of ways to access a parameter from the global data table, most of which are presented in the following code snippet:

```
'Methods of getting a Data Table value
val = DataTable.Value("ParamName",dtGlobalSheet)
val = DataTable.Value("ParamName","Global")

'By giving the sheet index starting from 1 for the global sheet
val = DataTable.Value("ParamName",1)

' Sheet name or id is a optional parameter and is assumed
' to be as for global data sheet in case not provided
val = DataTable.Value("ParamName")

' Value property is the default property of the DataTable object
' so DataTable("ParamName",dtGlobalSheet) is
' equivalent to DataTable.Value("ParamName",dtGlobalSheet)
val = DataTable("ParamName",dtGlobalSheet)
val = DataTable("ParamName")

'Using the data table object model
val = DataTable.GlobalSheet.GetParameter("ParamName").Value

'Using the data table object model
val = DataTable.GlobalSheet.GetParameter("ParamName").ValueByRow(1)
```

Problem 4-2. How to access a parameter from a Local data sheet

```
'Various methods to get data table value
val = DataTable.Value("ParamName",dtLocalSheet)
val = DataTable.Value("ParamName","<LocalActionName>")
val = DataTable("ParamName",dtLocalSheet)
Val = DataTable("ParamName","<LocalActionName>")

'The local sheet of the action which is executing this statement
val = DataTable.LocalSheet.GetParameter("ParamName").value
```

Problem 4-3. How to check if a Sheet exists

```
'Function to check if DataTable sheet exists
Function isSheetExists(sheetName)
   On error resume next
    isSheetExists = TRUE
    Err.clear
    Set objSheet= DataTable.GetSheet(sheetName)
    'In case error occured sheet does not exist
    If err.number<>0 then
         isSheetExists = FALSE
    End if
End Function
```

Problem 4-4. How to preserve format of data output to a data table

```
'This would be modified to 1.23456789E+010 due to auto formatting
DataTable("ParamName") = "12345678901"
'This will not be auto formatted and will be treated as text
DataTable("ParamName") = "'" & "12345678901"
```

Problem 4-5. How to check if a parameter exists in a specific sheet

```
'Check if a parameter exists in data table
Function isParameterExists(sheetName, paramName)
   On error resume next
    isParameterExists = TRUE
    Err.clear
    ParamTotal = DataTable.GetSheet(sheetName).GetParameter(paramName)
    'In case of error the parameter does not exist
    If err.number<>0 then
         isParameterExists = False
   End if
End Function
```

Problem 4-6. How to export contents of a WebTable to a data sheet. Let's assume that the first row of the data table contains the columns heading. We then add those as parameters of the data table:

```
'Variable declaration
Dim i,j
Dim rowCount,colCount
Dim cellText, objTable
```

```
'Get table object
Set  objTable = Browser("").Page("").WebTable("")

'Get the row count of the webtable
rowCount = objTable.RowCount

'Get the column count of the webtable header row
colCount = objTable.ColumnCount(1)

'create a output sheet
Set outSheet = DataTable.AddSheet ("Output")

'Create Parameters based on the 1st row of the web table
For i = 2 to colCount
   cellText = objTable.GetCellData(1,i)

   'Note in case the CellText contains space in between
   'then QTP will automatically convert it to a "_" character
   outSheet.AddParameter cellText,""
Next

'Skip first row as we assumed it to be a header row
For i = 2 to rowCount
   outSheet.SetCurrentRow i-1

   're-calculate the column count as some rows
   'have different column sizes
   colCount = objTable.ColumnCount(i)

   For j = 2 to colCount
         cellText = objTable.GetCellData(i,j)


         'We are using index here to avoid the problem of
         'the "_" issue if cell text has spaces or new line chars
         'then we will get an error. to overcome that we can also use
         'outSheet.GetParameter(Replace(cellText," ","_")).Value
         outSheet.GetParameter(j-1).value = cellText
   Next
Next
```

Problem 4-7. How to get value of a parameter from any specific row in the data table

We use the ValueByRow method to get value for any row

```
'Get a value by row
DataTable.GetSheet("SheetName").GetParameter("ParameterName").
ValueByRow(RowNumber)
```

Problem 4-8. How to execute a script for all Global Data Table iterations, when the script is set to run for only one iteration:

In case we want to manually repeat the code for each iteration, we need to write a bit code.

```
'Declare variable
Dim i, iCount

'Get the global sheet object
Set oGlobal = DataTable.GlobalSheet

'Get # of rows
iCount = oGlobal.GetRowCount

For i = 1 to iCount
   'Set the current row
   oGlobal.SetCurrentRow i
   'Execute the code to be repeated here
   Msgbox DataTable("UserName")
Next
```

Problem 4-9. How to get the number of columns that contain data:

To solve this problem we need to utilize the excel formula COUNTA. We add a parameter to the data table with the formula and then read its value:

```
'Add a new parameter with the formula
'For Columns 1 of data table use A1:A65536
'For column 2 of data table use B1:B65536 and so on
DataTable.GlobalSheet.AddParameter "New","=COUNTA(A1:A65536)"
'Get the new value
Msgbox DataTable("New")
```

NOTE: The above code won't work when there are no columns in the data table or all the columns have been used

Reader's Note

Chapter 16

Test Results

Test Results provides a summary of a QuickTest run session. The results are stored in an XML file format. QTP provides a Test Results viewer tool used to review and print these results. Test Results optionally contain image snapshots, based on QTP configuration settings. Figure 16-1 shows a typical test result summary:

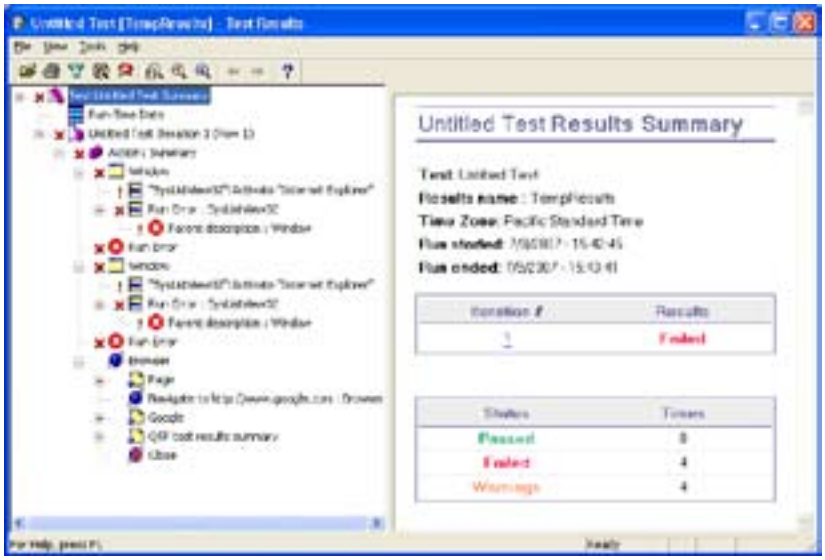


Figure 16-1. Test Results summary

The left hand pane provides summary information about each step performed during the test run. Icons to the left of each step provide the following information as well:

- A Cross (X) icon denotes a failed step
- An Exclamation (!) icon denotes a warning step
- A Tick (✓) icon denotes a passed step

- A step without any of these symbols denotes an information step

QTP assigns status to a step based on one of the follow situations:

- Checkpoint: Checkpoint can cause a step to pass or fail
- Smart Identification: If Smart identification is used to identify an object then that step is assigned the warning icon
- Error: If a step encounters any error, it will be assigned the failed icon
- Custom events: Custom events are used to directly assign a step an explicit status

Filtering Steps in a Report

It is possible to control what types of steps are written to the test results using the following statement:

```
Reporter.Filter = <Filter Value>
```

The <Filter Value> must use one of the following QTP built-in variables:

- rfEnableAll – Report all steps. This is the default setting
- rfEnableErrorsAndWarnings – Only report error (failed) and warning steps
- rfEnableErrorsOnly – Only report error steps
- rfDisableAll – Does not report any steps

The following code shows how to suppress a single checkpoint's pass/fail status:

```
'Store the old filter value
oldFilter = Reporter.Filter

'Disable reporting of all events
Reporter.Filter = rfDisableAll

Set oPg = Browser("Browser").Page("Page")
chkStatus = oPg.WebEdit("username").Check (Checkpoint("username"))

If chkStatus Then
  MsgBox "Passed"
Else
  MsgBox "Failed"
End If

'Restrore the old filter value
Reporter.Filter = oldFilter
```

Reporting Custom Steps

We can insert our own steps in the Test Results using the following statement:

```
Reporter.ReportEvent <EventStatus>, <ReportStepName>, <Details>
```

The <EventStatus> should use one of the following QTP built-in variables:

- micPass – Reports a step with passed status
- micFail – Reports a step with failed status
- micWarning – Reports a step with warning status
- micDone – Reports a step with no status

```
'Get the actual link href
actualLink = Browser("Browser").Page("Page").Link("Login").
GetROProperty("href")

If actualLink = "http://mywebsite.com/login.do" Then
  Reporter.ReportEvent micPass, "Validate Link - Login", "Correct Link"
Else
  Reporter.ReportEvent micFail, "Validate Login", "Wrong Link - " & actualLink
End if
```

While the <ReportStepName> and <Details> parameters are plain text strings, it is possible to embed HTML tags into these strings as follows:

```
'HTML text to be entered
sHTML = "&lt;<A target=_New href=""http://www.mywebsite.com"">Click Me</A>&gt;"

'Add to reporter
Reporter.ReportEvent micDone, "Link", sHTML
```

QTP also supports one more undocumented EventStatus, micInfo. Using micInfo creates a step with an "i" icon for the step. This is useful to report just information in the report, which we may want to visually segregate from the similar micDone entries.

Inserting Files in Test Results

Consider the following code:

```
'Create the html file path
'store it in a Test Results folder
sFile = Reporter.ReportPath & "\StepsToRecreate.html"

'Create the HTML file
Set FSO = CreateObject("Scripting.FileSystemObject")
Set file = FSO.CreateTextFile(sFile,True)
file.Write "<I><B>Step 1</B></I>. Launch www.mywebsite.com"
file.Write "<P><B>Step 2</B>. Click on the Login link"
file.Write "<P><B>Step 3</B>. Enter the username as tarun"
```

```
file.Write "<P><B>Step 4</B>. Enter password as tarun"
file.Write "<P><B>Step 5</B>. Click on the Login button"
file.Write "<P><B><I>Step 6</I></B>. Close the browser"
file.close

'Insert the above file as a IFRAME in the report
sHTML = "&lt;<IFRAME width=""100%"" height=250 src=""file:///" &
sFile & """></IFRAME>&gt;"
Reporter.ReportEvent micPass, "Steps To Recreate", sHTML

'Clean up
Set file = Nothing
Set FSO = Nothing
```

Running the above code will display the file as shown below:

Figure 16-2. Embedding File in test results

Inserting Snapshots in Test Results

This section describes various ways of inserting screen snapshots into the Test Results.

Method 1

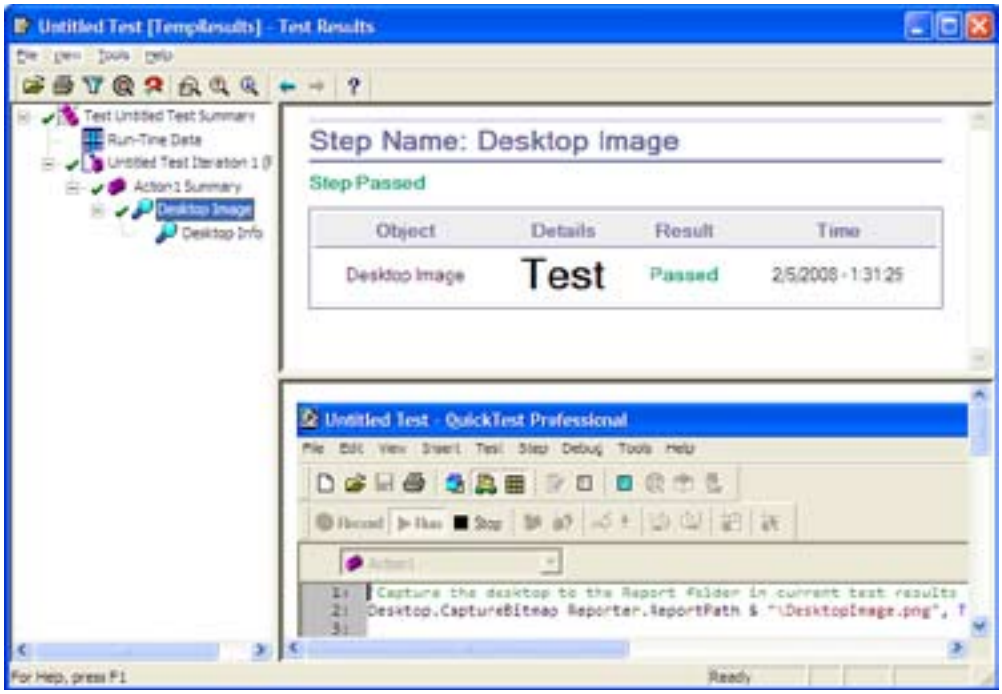
Configure QTP to save a screen snapshot for every step. Go to *Tools→Options…→Run (Tab)* and set the option for *"Save step screen capture to results:"* **to** *"Always"* as shown in the Figure 16-3.

These techniques have the following advantages over the ReportEvent method:

- Support for relative paths
- Inserts HTML step information without having to use "<" and ">", etc. HTML tags.

The Test Results produced using the script above is shown in the Figure 16-4

Figure 16-4. Custom Test results



NOTE: The images inserted using above method are also visible when results are viewed from Quality Center. This was considered a impossible task till QTP 10

Accessing Test Results at the end

QTP creates the results in the test results folder. We may want to access these results at the end of the test script to save them to another location or to send them through email. Let's say we want to copy the Results.xml file which is created in the Report folder. We can write the following code to perform this task in a QTP script:

```
'Report a pass event
Reporter.ReportEvent micPass, "Testing Report", "Testing Exporting of Report"

'Get the result directory
sResultDir = Environment("ResultDir")

'Copy the file to destination
Set fso = CreateObject("Scripting.FileSystemObject")
fso.CopyFile  sResultDir  & "\Report\Results.xml", "C:\Copy_Results.xml", True
Set fso = Nothing
```

Appendix A: Problems discussed in the book

Appendix B: Index

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

HP QuickTest Professional is a functional test automation tool. It supports a Record and Playback framework out of the box, where we can record and capture our interactions with the application under test and then replay those actions later. With this book you will learn

- Basic concepts of QTP
- Working without Object repository using Descriptive Programming
- Advanced concepts of QTP
- Working with external tools Microsoft Word, Outlook, Excel
- Integrating QTP Scripts with Quality Center
- Real life Automation problems and their solutions

Quotes from reviewers

"I find this to be a very pragmatic, hand's on book for those who want to extend their QTP skills beyond basic expert view programming. This book is written by a QTP master for those who wish to eventually become masters themselves."

– Terry

"Tarun Lalwani has singlehandedly helped thousands of people to expand their knowledge of QuickTest Professional. Here is a book the automated testing community has been crying-out for. This book will help QTP practitioners, from beginner to expert. I have used QTP from V6.0 and during the review I learnt something from every chapter."

– Mark

About the Author

Tarun works as an Automation & Solution consultant and has over 5 years of experience in IT. He has worked on various Automation projects using QTP, VBScript, VB6, VB.NET, C#.NET, Excel Macro, Outlook Macro. He blogs at KnowledgeInbox.com.
He can be contacted through http://KnowledgeInbox.com/contact-me

For support & queries please use http://KnowledgeInbox.com/forums/