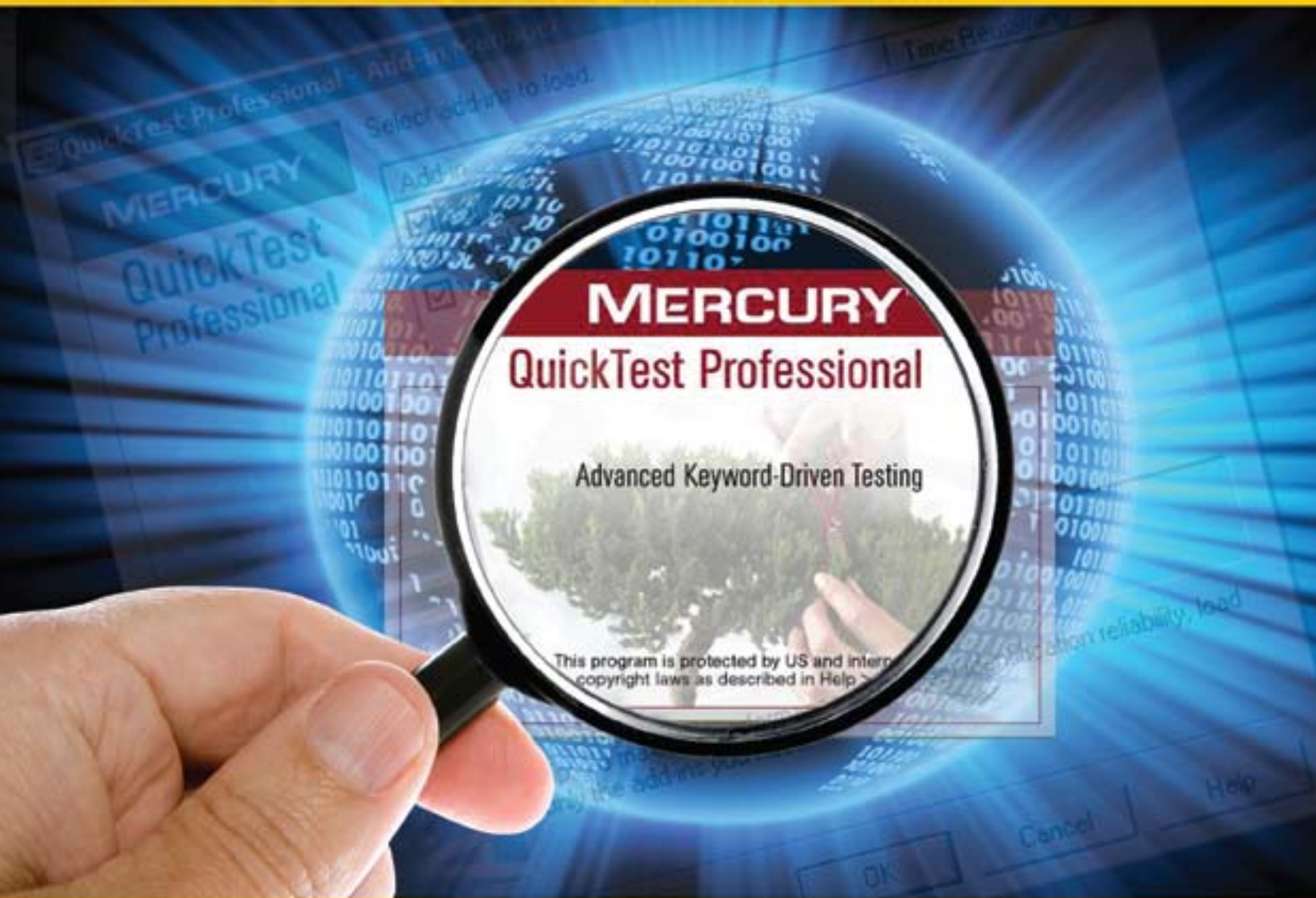


# QuickTest Professional Unplugged



**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

---

<b>Preface.</b>	<b>xii</b>
<b>1. Introduction</b>	<b>1</b>
What is Test Automation?	1
When Should Test Automation Be Used?	1
When Should Test Automation Be Avoided?	2
The Automated Testing Process.	2
What is HP QuickTest Professional (QTP)	4
<b>2. Using QTP Help</b>	<b>7</b>
Contents Tab	8
Properties returned by the GetROProperty Function.	8
Methods provided by a Test Object	8
Index Tab	9
Search Tab	10
<b>3. Object Repository (OR).</b>	<b>13</b>
How objects are added to the OR	14
Test and Run-time Objects.	16
TO Properties.	16
Object Repository Modes	18
Per-Action versus Shared Object Repository.	18
The Object Spy	19
Object Identification.	21
User-defined Objects	22
Object Repository Shortcomings	23
<b>4. DataTables</b>	<b>27</b>
Design and run-time data table	29
Design time data table	29
Run-time data table	29
When to use the global or a local data table	29
Setting data table iterations	30
Data table object model	32

Data table formatting . . . . .	32
<b>5. Actions. . . . .</b>	<b>37</b>
Input and Output Parameters . . . . .	37
Types of Actions . . . . .	40
Inserting Calls to Actions. . . . .	40
Insert Call to New... . . . .	41
Insert Call to Existing... . . . .	42
Insert Call to Copy... . . . .	43
<b>6. QTP Environment Variables . . . . .</b>	<b>45</b>
Types of Environment variables . . . . .	45
Environment Object. . . . .	46
<b>7. Utility Objects. . . . .</b>	<b>55</b>
The Crypt Object . . . . .	55
The OptionalStep Object . . . . .	56
The PathFinder Object. . . . .	56
The RandomNumber Object . . . . .	57
The Setting Object . . . . .	57
The WebUtil Object. . . . .	59
The SystemUtil Object. . . . .	59
The RegisterUserFunc Object . . . . .	60
<b>8. Checkpoints . . . . .</b>	<b>63</b>
Types of checkpoints . . . . .	63
Problems with QTP built-in checkpoints. . . . .	64
Alternatives to QTP checkpoints . . . . .	65
Updating Checkpoints at run-time. . . . .	65
<b>9. Library Files . . . . .</b>	<b>69</b>
Associating a Library globally with a Test . . . . .	69
Dynamically Loading libraries locally at run-time . . . . .	70
Dynamically Loading libraries globally at run-time . . . . .	70
Understanding Execution Scope . . . . .	72
Applicability of Option Explicit . . . . .	73
Executing code in Local scope from within Global Scope . . . . .	73
<b>10. Descriptive Programming (DP) . . . . .</b>	<b>77</b>
Object Identification. . . . .	77
Implicit Properties . . . . .	78
When to use Descriptive Programming. . . . .	78
Descriptive Programming Syntax . . . . .	78
Using description objects. . . . .	78
The micclass Identifier. . . . .	80
Using Description Strings . . . . .	80



Enumerating ChildObjects. . . . .	80
Converting an OR-based script to a DP-based script. . . . .	81
Using a VBScript DP object repository . . . . .	83
Problems with Ordinal Identifiers . . . . .	84
Working with Multiple Browser . . . . .	86
Browser identification Issue. . . . .	87
Browser identification using OpenTitle . . . . .	87
Browser Identification using a unique OpenURL Property . . . . .	88
<b>11. Debugging in QTP . . . . .</b>	<b>91</b>
Establishing our Debugging Configuration . . . . .	91
Using Breakpoints . . . . .	92
Working with the Watch Expression Tab. . . . .	92
Working with the Variables Tab . . . . .	93
Working with the Command Tab . . . . .	94
<b>12. Recovery Scenarios . . . . .</b>	<b>97</b>
When not to use recovery scenarios? . . . . .	97
Situations in which a Recovery Scenario won't work . . . . .	101
Recovery Scenario for IE Authentication window . . . . .	103
Default Recovery Scenarios. . . . .	104
The Recovery object . . . . .	105
Recovery Function Calls . . . . .	105
Error Handling . . . . .	107
<b>13. Regular Expressions . . . . .</b>	<b>111</b>
Regular Expression Characters . . . . .	112
When to use Regular Expressions in QTP. . . . .	115
Testing a Regular Expression Using Code. . . . .	117
Using a Regular Expression to Select a WebList or WebRadioGroup Value . . . . .	118
Extracting Information Using a Regular Expression. . . . .	118
Replacing Data from a String Using a Regular Expression. . . . .	119
<b>14. VBScript . . . . .</b>	<b>123</b>
Strings . . . . .	123
Conversion function. . . . .	131
Date and Time functions . . . . .	132
Misc . . . . .	135
TypeName . . . . .	135
VarType . . . . .	135
GetRef . . . . .	135
CreateObject . . . . .	136
GetObject. . . . .	136
Working with Arrays . . . . .	142
Fixed length arrays. . . . .	142

Dynamic Arrays . . . . .	142
Execute Statement . . . . .	143
Eval function . . . . .	144
Optional arguments in VBScript . . . . .	144
<b>15. Synchronization . . . . .</b>	<b>151</b>
Synchronization Points . . . . .	151
1st Method – Using the Sync method . . . . .	151
2nd Method – Using the Wait statement . . . . .	152
3rd Method – WaitProperty method. . . . .	152
Checking Object Existence . . . . .	152
Browser Vs Page Sync. . . . .	153
Events. . . . .	153
Replay Options. . . . .	155
<b>16. Test Results. . . . .</b>	<b>159</b>
Filtering Steps in a Report . . . . .	160
Reporting Custom Steps . . . . .	161
Inserting Files in Test Results . . . . .	161
Inserting Snapshots in Test Results . . . . .	162
Converting XML Results to HTML. . . . .	164
Configuring QTP to generate HTML results . . . . .	164
The QTP Reporter Object . . . . .	165
Accessing Test Results at the end . . . . .	167
<b>17. Working with APIs. . . . .</b>	<b>171</b>
Extern object . . . . .	171
VB API Definition Syntax . . . . .	171
QTP API Definition . . . . .	172
QTP API Limitation. . . . .	176
API using COM Objects in VB6 . . . . .	176
Running API using Excel. . . . .	179
Dynamically generating an Excel Macro. . . . .	181
Working with Modal dialog boxes. . . . .	184
<b>18. QTP Automation Object Model . . . . .</b>	<b>189</b>
QTP Automation Object Model (AOM) . . . . .	189
Passing arguments to Test Cases . . . . .	190
<b>19. HTML DOM . . . . .</b>	<b>195</b>
When to use the HTML DOM. . . . .	195
When not to use the HTML DOM. . . . .	196
HTML DOM Objects. . . . .	196
HTML Node. . . . .	196
The Document Object . . . . .	197
The HTML Element Object. . . . .	197

The HTML Element Collection . . . . .	197
Getting the Web page Element using the DOM . . . . .	198
Other HTML Elements . . . . .	199
Converting a DOM object to QTP Test Object . . . . .	202
Checking Appearance using Style Sheets . . . . .	202
Checking Visibility of a Object on a Page . . . . .	202
Selecting a Radio button by Text . . . . .	204
<b>20. Working with Web Tables . . . . .</b>	<b>211</b>
Web Tables . . . . .	211
Accessing the WebTable . . . . .	212
Using Index . . . . .	213
Using Name or HTML ID . . . . .	213
Using innerText/outerText . . . . .	213
Using OR . . . . .	216
Using an object inside the table . . . . .	218
Getting the location of an object inside the table . . . . .	219
Clicking inside a WebTable . . . . .	221
Asynchronous Table loading . . . . .	223
Exporting WebTable to a DataTable . . . . .	223
Extending WebTable functionality . . . . .	224
Getting a QTP Table from a element . . . . .	224
Finding a cell in a Table . . . . .	225
<b>21. Working with XML . . . . .</b>	<b>231</b>
QTP XML Objects . . . . .	232
Reading and Modifying XML . . . . .	232
Working with Multiple environments of an Application . . . . .	235
Exporting XML to a DataTable . . . . .	237
Comparing XML . . . . .	238
<b>22. Working with Databases . . . . .</b>	<b>243</b>
Connecting to a Database . . . . .	243
Building Connection Strings . . . . .	244
Checking the State of a Connection or Recordset . . . . .	247
Exporting the Results of a Query to Data Table . . . . .	248
Executing a Stored DB procedure . . . . .	249
<b>23. Working with Microsoft Excel . . . . .</b>	<b>253</b>
The Excel Automation Object Model . . . . .	253
Converting VBA to VBScript . . . . .	258
Performance Enhancements . . . . .	264
<b>24. Working with Microsoft Outlook . . . . .</b>	<b>269</b>
Outlook Object model . . . . .	269
Launching the Outlook Application . . . . .	270



Logging in to the Namespace . . . . .	270
Enumerating the top level folder . . . . .	270
Getting a Folder interactively from a user . . . . .	272
Getting the default folders . . . . .	272
Reading e-mail Messages . . . . .	273
Accessing e-mail Message Properties . . . . .	274
Downloading attachments . . . . .	275
Sending an E-mail Message . . . . .	275
Outlook Security Dialogs . . . . .	276
Getting addresses from e-mail messages . . . . .	279
Extended Email Address . . . . .	279
Clicking links inside an e-mail message . . . . .	280
Launching QTP using an email trigger . . . . .	281
<b>25. Working with Microsoft Word . . . . .</b>	<b>285</b>
Word Automation Object Model . . . . .	285
<b>26. Working with Microsoft Internet Explorer . . . . .</b>	<b>295</b>
Launching Internet Explorer (IE). . . . .	295
IE COM Methods and Properties . . . . .	296
Page Synchronization . . . . .	297
Enumerating all IE Windows . . . . .	298
Finding an IE window . . . . .	299
Launching a new browser . . . . .	299
Getting Web page DOM . . . . .	301
Accessing webpage script variables . . . . .	301
Using IE to get user input . . . . .	302
IE Popup Dialogs . . . . .	304
Disabling IE dialogs . . . . .	304
Changing IE settings using the Registry . . . . .	304
Popup Blocker . . . . .	305
Disable Script Error dialog . . . . .	305
Security Alert – Redirection popup . . . . .	306
Security Alert – Certificate warning . . . . .	307
Security Alert – Secure Connection . . . . .	307
Security Information – Secure and non-secure item . . . . .	308
Active content/Java Script Prompt . . . . .	309
File download – Information bar . . . . .	309
Handling popup dialog by using a code . . . . .	310
File Download – Security Warning popup . . . . .	311
Checking for Broken images on a Web Page . . . . .	314
Using a Browser Object as a Window Object . . . . .	314
Custom Browser Applications . . . . .	316

<b>27. Working with HP Quality Center.....</b>	<b>319</b>
Quality Center .....	319
Connecting QTP with QC .....	319
QC Paths .....	321
Relative Paths in QC .....	321
Resolving Relative Paths .....	321
QCUtil Object .....	322
QC Open Test Architecture (OTA) .....	324
The TDConnection Object.....	324
The Command and Recordset Objects.....	327
The AttachmentFactory Collection .....	328
Simple way of Downloading files from QC .....	330
Uploading Attachments to QC.....	331
Getting the Current Test Location .....	332
Enumerating All Tests present in a TestLab folder .....	332
Enumerating all the Tests in a Test Lab Tab .....	334
Getting the Current Test Set Location .....	335
<b>28. Advanced QTP .....</b>	<b>337</b>
Synchronizing Script Execution between different machines .....	337
Enumerating the Setting Variables.....	338
Setting Persistence .....	343
Stop and Re-Run QTP using Code .....	344
Adding Recovery Scenarios at Run-time.....	345
Executing code when Script ends .....	346
Making an Object visible on a Web Page .....	348
Advanced Text Checkpoints .....	348
Extending Test Objects using Classes .....	355
Using JScript in QTP .....	357
Passing variable number of arguments to a function.....	358
Scope differences .....	358
Using Try...Catch block .....	359
Working with JScript Arrays .....	360
Working with JS classes.....	360
Object Class Mapping .....	361
<b>29. What's New in QTP 9.2.....</b>	<b>365</b>
IDE Enhancement .....	365
Object Spy .....	367
Web Drag and Drop Support .....	368
Menu updates .....	368
New Configuration Options.....	369
Object Repository Enhancements .....	370

Added and Enhanced Utility Objects .....	370
The Print Log Utility .....	372
Hiding the Print Window .....	372
Showing the Print window .....	373
Clearing the Print Window .....	373
Programmatically Capturing the Print Log text .....	374
micRegExpMatch .....	375
MercuryTimers .....	375
RepositoriesCollection .....	377
DotNetFactory .....	377
Generating Object name from the object .....	378
<b>30. Working with .NET Classes .....</b>	<b>381</b>
Using the DotNetFactory Object .....	381
Passing Parameters to Class Constructors .....	381
Passing Parameters to .NET Objects .....	382
Passing Enums .....	383
Working with .NET Arrays .....	384
Playing a Wav File .....	385
Working with the Clipboard .....	385
Getting Computer Information .....	385
Accessing the Registry .....	386
Ping an IP Address .....	386
Evaluating Keyboard Control Key Status .....	386
.Net Arrays Revisited .....	387
Working with .Net Stacks .....	387
Working with .Net Queues .....	388
Working with .Net Date and Time Formatting .....	388
Sending Emails using .NET .....	389
Converting Images to other File Formats .....	390
Getting user Input using .NET Forms .....	392
<b>31. Designing Frameworks .....</b>	<b>399</b>
Design guidelines .....	399
Support of different application versions .....	400
QTP Examples .....	402
Reusability .....	402
Support for different application versions .....	403
Externally Configurable .....	404
Logging .....	404
Self configurable .....	405
<b>32. Useful Tools .....</b>	<b>409</b>
QTP Script Editor .....	409

VBSEdit . . . . .	410
Notepad++ . . . . .	410
QTP Uninstaller . . . . .	411
ScreenCapture API . . . . .	412
KnowledgeInbox IE Session Cookies Helper . . . . .	413
VB 2 QTP API Converter . . . . .	413
IE WebDeveloper . . . . .	413
Test Design Studio (TDS) . . . . .	415
RAD Regular Expression Designer . . . . .	416
<b>Appendix A: Problems discussed in the book . . . . .</b>	<b>A-I</b>
<b>Appendix B: Index . . . . .</b>	<b>B-I</b>

# 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](http://www.SQAForum.com), without knowing that I will specialize in the use of this tool in future. After sometime I launched [KnowledgeInbox.com](http://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 then 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](mailto: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](mailto:thorwath@lakefolsom.com).

Siddharth has worked as an 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](mailto: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, hands 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.

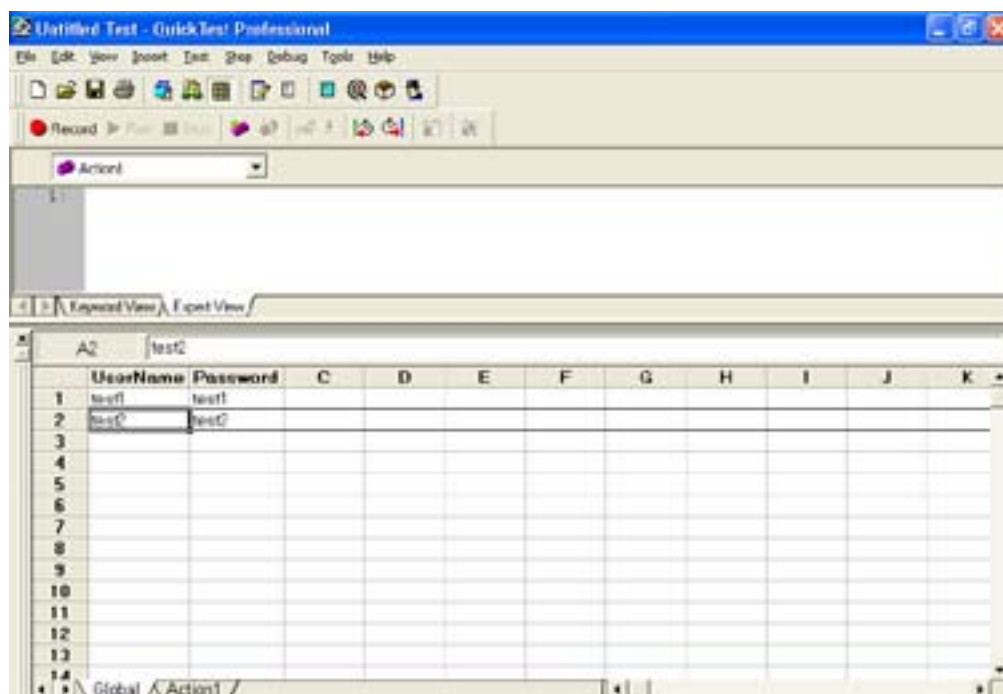


Figure 4-1. DataTable

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

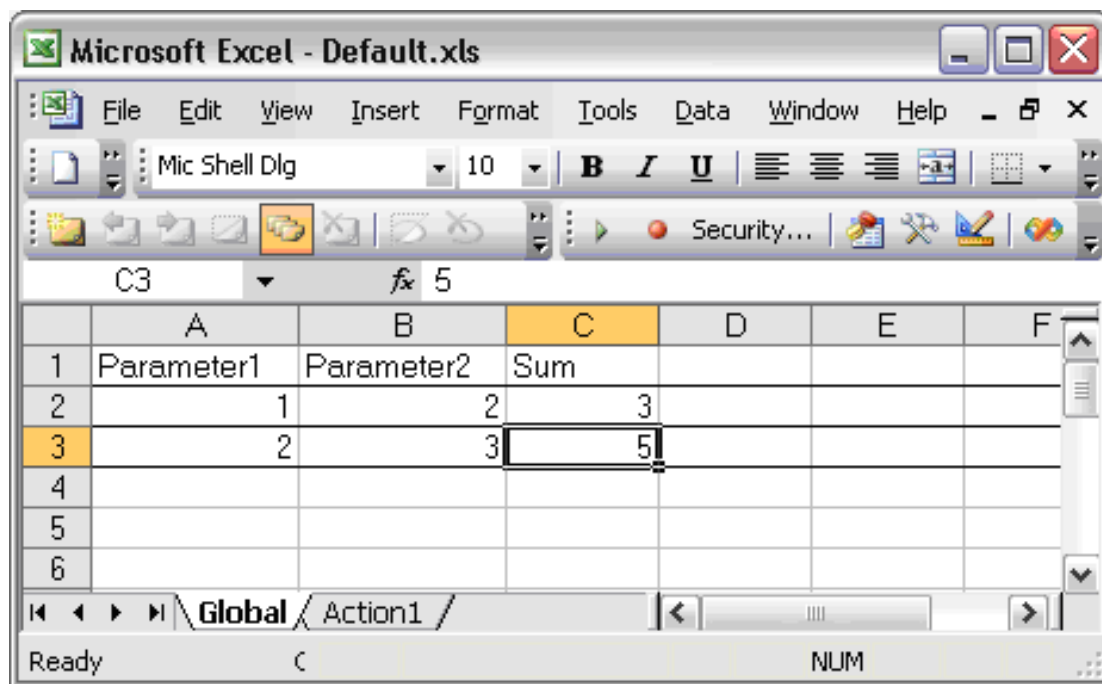


Figure 4-2. Default.xls Data table

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.

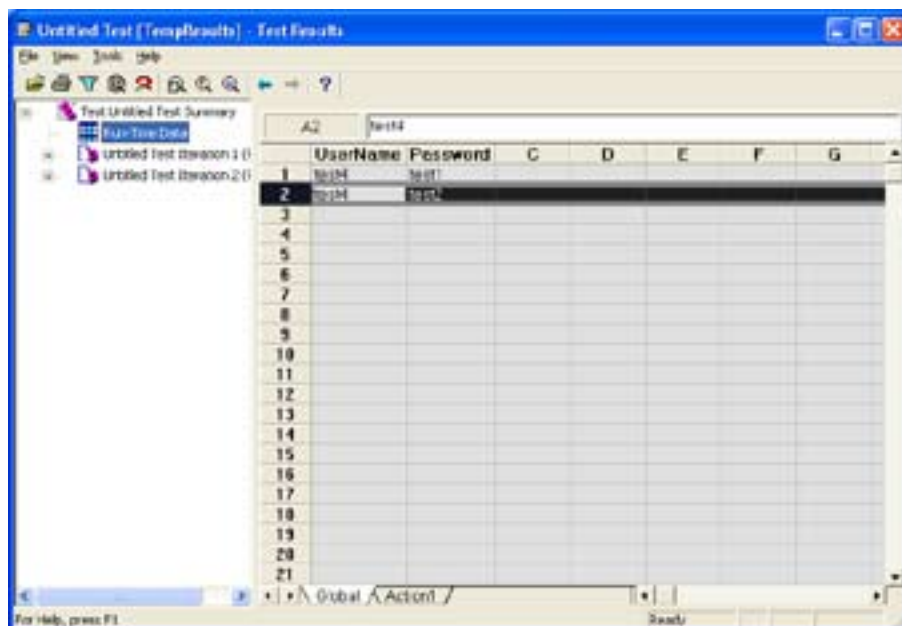
# Design and run-time data table

## 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



	A2	B	C	D	E	F	G
1		Test1	Test1				
2		Test2	Test2				
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							

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.

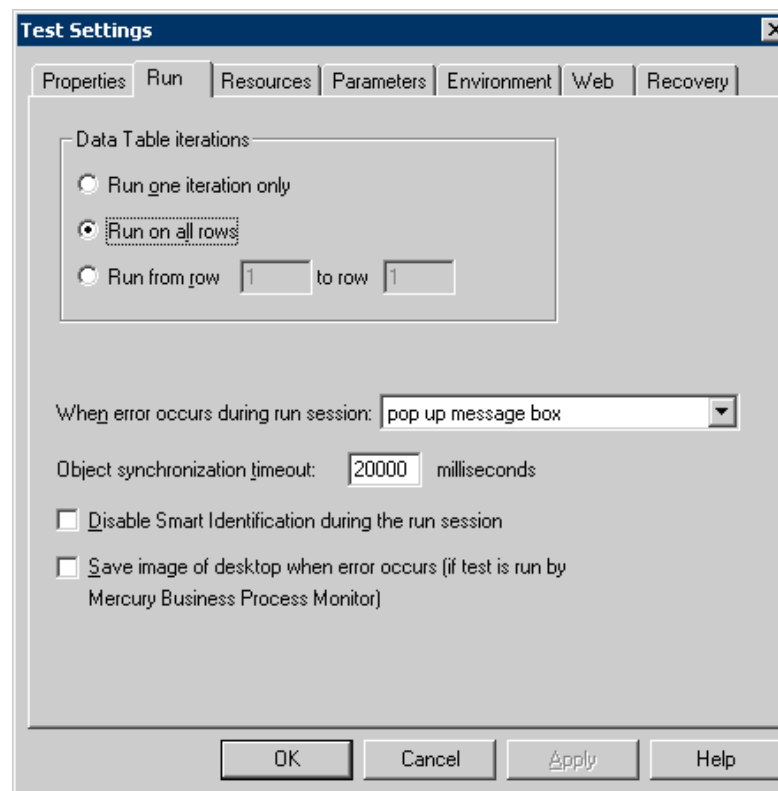


Figure 4-4. Global data iterations



**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

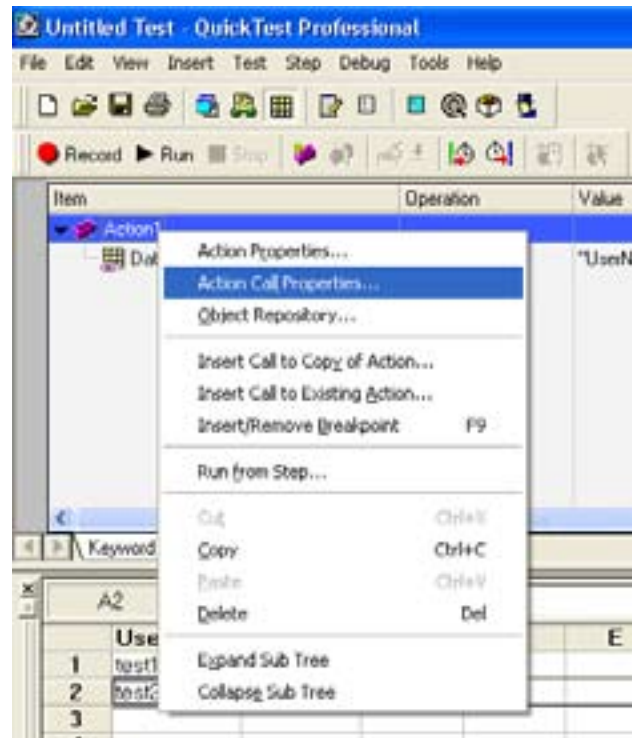


Figure 4-5. Action Call Properties - Keyword View

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

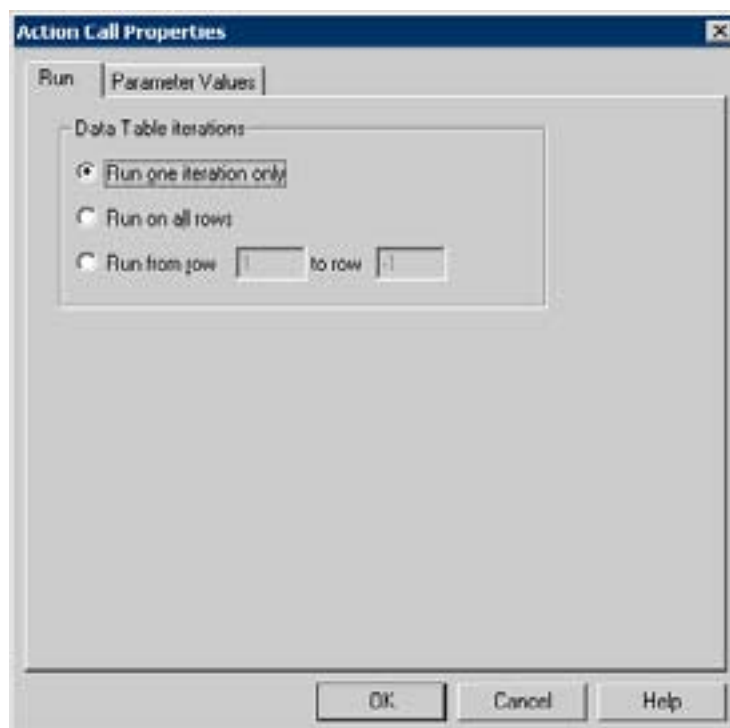


Figure 4-6. Action Call Properties dialog

## 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 occurred 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.AddParameter(Replace(cellText," ","-")).Value
        outSheet.AddParameter(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:

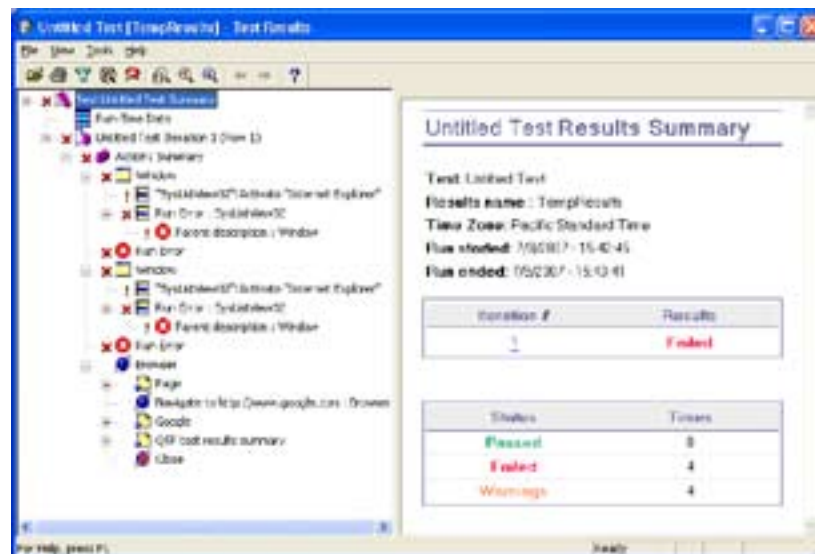


Figure 16-1. Test Results summary

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

'Restore 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;&lt;A target=_New href=""http://www.mywebsite.com""&gt;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 "&lt;I&gt;&lt;B&gt;Step 1&lt;/B&gt;&lt;/I&gt;. Launch www.mywebsite.com"
file.Write "&lt;P&gt;&lt;B&gt;Step 2&lt;/B&gt;. Click on the Login link"
file.Write "&lt;P&gt;&lt;B&gt;Step 3&lt;/B&gt;. 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 FS0 = 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.

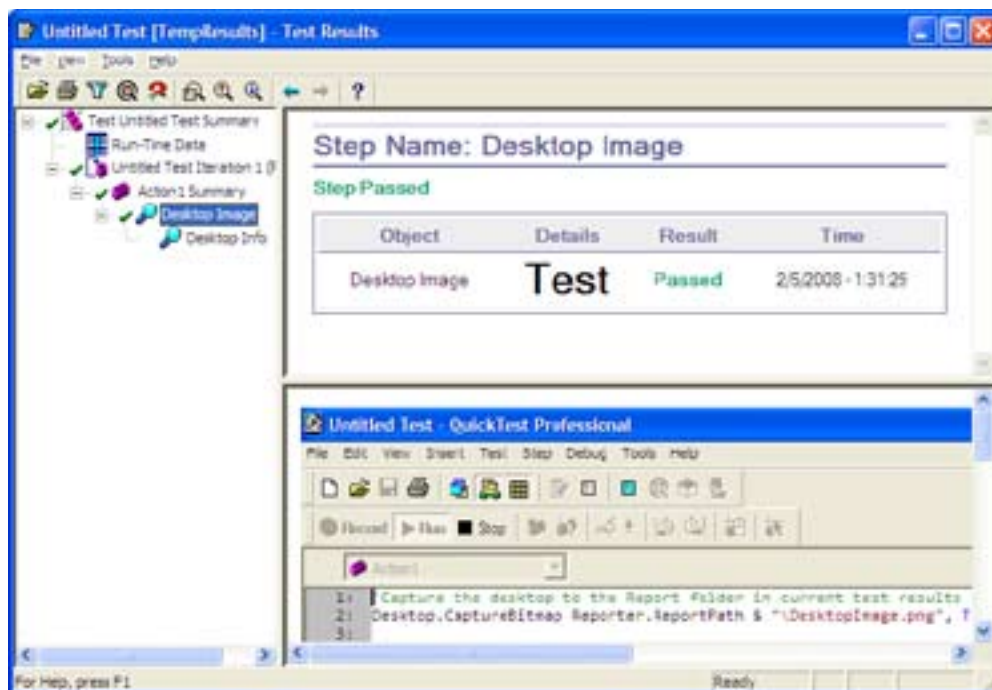


These techniques have the following advantages over the ReportEvent method:

- Support for relative paths
- Inserts HTML step information without having to use “&lt;” and “&gt;”, 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

---

<b>Problem 3-1.</b> Working with Test Object properties .....	16
<b>Problem 3-2.</b> Changing Test Object properties at run time .....	17
<b>Problem 3-3.</b> Getting Run-time Object properties during test execution .....	17
<b>Problem 4-1.</b> How to access a parameter from the global data sheet .....	32
<b>Problem 4-2.</b> How to access a parameter from a Local data sheet .....	33
<b>Problem 4-3.</b> How to check if a Sheet exists .....	33
<b>Problem 4-4.</b> How to preserve format of data output to a data table .....	33
<b>Problem 4-5.</b> How to check if a parameter exists in a specific sheet .....	33
<b>Problem 4-6.</b> 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: .....	33
<b>Problem 4-7.</b> How to get value of a parameter from any specific row in the data table. ....	34
<b>Problem 4-8.</b> How to execute a script for all Global Data Table iterations, when the script is set to run for only one iteration: .....	35
<b>Problem 4-9.</b> How to get the number of columns that contain data: .....	35
<b>Problem 6-1.</b> How to access an environment variable? .....	47
<b>Problem 6-2.</b> How can we check if an environment variable exists? .....	47
<b>Problem 6-3.</b> How is an environment variable defined at run time? .....	47
<b>Problem 6-4.</b> How can environment variables be exported to an XML file? .....	48
<b>Problem 6-5.</b> How are QTP record and run settings changed using Environment variables? .....	49

<b>Problem 6-6.</b> How is an environment variable cleared or destroyed? .....	51
<b>Problem 6-7.</b> How an object is passed using an environment variable? .....	51
<b>Problem 6-8.</b> How an array is passed using Environment variables? .....	52
<b>Problem 6-9.</b> How are environment variables loaded from an external XML file? .....	52
<b>Problem 8-1.</b> How can we check if a checkpoint passed or failed? .....	64
<b>Problem 9-1.</b> How are global variables defined at run-time? .....	72
<b>Problem 12-1.</b> How can we get the status of an Action as Pass/Fail at the end of the action? .....	102
<b>Problem 13-1.</b> What will the regular expression be for a domain which can have possible value of “ test.app.com”, “qa.app.com”, “www.app.com” .....	114
<b>Problem 13-2.</b> What will be the r.e. matching a date in the format MM/DD/YY? .....	114
<b>Problem 14-1.</b> How can we find the current page and the total page values from the string “Page 4 of 15”? .....	125
<b>Problem 14-2.</b> How can we count the number of “if” phrases in the “if + if = 2 if” string? .....	125
<b>Problem 14-3.</b> Given a complete file path specification, extract the filename, file path and drive letter from the specification .....	127
<b>Problem 14-4.</b> Let’s revisit the page number extraction problem again. How can we find the current page and the total page values from the string “Page 4 of 15”? .....	128
<b>Problem 14-5.</b> How can we split a string into words or lines? .....	129
<b>Problem 14-6.</b> How can the number of business/working days be calculated between two dates (i.e. difference should exclude the weekends)? .....	133
<b>Problem 14-7.</b> Get a sub-string in a string between two specified characters. ....	136
<b>Problem 14-8.</b> Function to get a substring from a start character in a string. ....	137
<b>Problem 14-9.</b> Write a function that reads a string word by word where two words would be separated by one or more spaces.....	138
<b>Problem 14-10.</b> Create a function TrimAll which trims not only spaces but other whitespace characters like new lines, carriage returns, and line feeds etc.: .....	138
<b>Problem 14-11.</b> Create a FilterStr function which takes a string and character set as input and will filter the string based on the characters in the character set. ....	139
<b>Problem 14-12.</b> Create two functions, one which returns the path portion and another the file name, when passed a fully qualified path or filename: .....	140
<b>Problem 14-13.</b> Write a function that compares two string ignoring new line, line feeds, tab etc. whitespace characters: .....	141

<b>Problem 14-14.</b> Write a function that replaces repeated occurrences of a given character by another character. ....	141
<b>Problem 14-15.</b> How can we get the UBound/LBound values of multi-dimensional array? .....	143
<b>Problem 14-16.</b> How can we declare and initialize a read only variable at run-time? ....	144
<b>Problem 17-1.</b> How can we determine if the topmost window on the desktop is a browser window? .....	172
<b>Problem 17-2.</b> How can we get the value of a Windows Environment variable? (Note that QTP environment variables differ from Windows environment variables)....	172
<b>Problem 17-3.</b> How can we check (select) an item in a List box using a Windows API? ..	172
<b>Problem 17-4.</b> How can we get the background color of a text box (helpful when a color for a mandatory field is different than an optional color)? .....	173
<b>Problem 17-5.</b> How can we simulate a keyboard event using the Windows APIs? .....	173
<b>Problem 17-6.</b> How can we prevent a PC from being locked by its screen saver? .....	175
<b>Problem 17-7.</b> How can we maximize a window or a browser? .....	176
<b>Problem 17-8.</b> How can we download a file from a URL to disk? .....	176
<b>Problem 18-1.</b> How can we use AOM to run a QTP test case? .....	189
<b>Problem 18-2.</b> How can we use AOM to change the starting and ending iterations for a test case? .....	190
<b>Problem 18-3.</b> How can we use AOM to launch QTP on a remote machine? .....	192
<b>Problem 21-1.</b> How can we copy an XML file and save it to another XML file? .....	232
<b>Problem 21-2.</b> How can we get an entire XML file into a string and then save it to another XML file using that string? .....	233
<b>Problem 21-3.</b> How can we create the following XML code at run-time in QTP, and then save it to an XML file? .....	233
<b>Problem 21-4.</b> How can we load environment variables from multiple XML files? .....	234
<b>Problem 21-5.</b> How can we modify the contents of an XML file? .....	236
<b>Problem 21-6.</b> How can we export a part of XML to a DataTable? .....	237
<b>Problem 21-7.</b> How can we extract XML from a URL or Web browser? .....	238
<b>Problem 21-8.</b> How can we compare two XML documents for equality? .....	238
<b>Problem 21-9.</b> How can we compare two xml file with different ordering of nodes? ....	238
<b>Problem 22-1.</b> How can we execute a query to access all records? .....	246

<b>Problem 22-2.</b> How can we determine the number of rows altered by an update or a delete query? .....	248
<b>Problem 23-1.</b> How can we invoke and terminate an instance of Excel? .....	254
<b>Problem 23-2.</b> How can we create a new excel sheet and then save it? .....	254
<b>Problem 23-3.</b> How can we determine if an Excel file exists, and if not, how to create one? .....	255
<b>Problem 23-4.</b> How can we access all non-blank row values in a specific column? .....	256
<b>Problem 23-5.</b> How can we search for a text string and then update another cell's value in that row? .....	256
<b>Problem 23-6.</b> How can we use an Excel sheet as a QTP output Data Table? .....	257
<b>Problem 23-7.</b> How can we allow a user to select a file to be opened? .....	258
<b>Problem 23-8.</b> How can we access an Excel spreadsheet embedded in Internet Explorer? .....	262
<b>Problem 25-1.</b> How to instantiate and terminate a word application? .....	286
<b>Problem 25-2.</b> How to get reference to an already open word application? .....	286
<b>Problem 25-3.</b> How to enumerate all the open word documents? .....	286
<b>Problem 25-4.</b> How to open and save a new document in word? .....	287
<b>Problem 25-5.</b> How to get reference to an already existing word document? .....	287
<b>Problem 25-6.</b> How to open a word document in the read only mode? .....	288
<b>Problem 25-7.</b> How to print an open word document? .....	288
<b>Problem 25-8.</b> How to insert text into a word document from a pre-defined style? .....	288
<b>Problem 25-9.</b> How to insert an image in a word document and scale its size by 50%? .....	289
<b>Problem 25-10.</b> How to find and replace some text in a word document? .....	289
<b>Problem 25-11.</b> How to insert a table into a word document? .....	290
<b>Problem 25-12.</b> How to change the font for the text being written in a word document? .....	291
<b>Problem 25-13.</b> How to preserve the text already present in a word document while writing? .....	292

# Appendix B: Index

---

## A

- Actions 37
  - Input and Output Parameters 37
  - Insert Call to Copy... 43
  - Insert Call to Existing... 42
  - Insert Call to New... 41
- actionsspath 343
- AddAttachment 331
- AddParameter 34–35
- AddSheet 34
- ADODB.Command 243, 247, 249
- ADODB.Connection 243
- ADODB.Recordset 243, 246
- Advanced Text Checkpoints 348
- API using COM Objects in VB6 176
- API using Excel 179
- ApplyXSL 239
- Array 128
- Asc 131
- Automation Object Model (AOM) 189

## B

- Browser Vs Page Sync 153

## C

- CBool 131
- CByte 131
- CCur 131
- CDate 131
- Check 64
- CheckBrokenImages 314
- Checkpoints 63
  - Alternatives to QTP checkpoints 65
  - Types of checkpoints 63
  - Updating Checkpoints at run-time 65

- ChildObjects 80
- Chr 132
- CLng 131
- CleanBrowserPopups 310
- CleanText 141
- CLng 131
- CloseDescendentProcesses 59
- CloseProcessByHwnd 59
- CloseProcessById 59
- CloseProcessByName 59
- CloseProcessByWndTitle 59
- Comparing XML 238
- Converting a DOM object to QTP Test Object 202
- Converting VBA to VBScript 258
- CreateDescription 79
- CreateObject 136
- CreationTime 84
- Crypt Object 55
- CSng 131
- CStr 131
- CurrentRun 323
- currentStyle 203
- CurrentTest 323
- CurrentTestSet 323
- CurrentTestSetTest 323

## D

- Database 243
- DataTable 27
  - Data table formatting 32
  - Data table object model 32
  - Design time data table 29
  - Run-time data table 29
  - Setting data table iterations 30
  - When to use the global or a local data table 29

- Date 132
- DateDiff 133
- Day 132
- Debugging 91
  - Using Breakpoints 92
  - Working with the Command Tab 94
  - Working with the Variables Tab 93
  - Working with the Watch Expression Tab 92
- DefaultObjRepType 343
- DefaultTSRFilePath 343
- Descriptive programming 77
  - Browser identification Issue 87
  - Browser Identification using a unique OpenURL Property 88
  - Browser identification using OpenTitle 87
  - Converting an OR-based script to a DP-based script 81
  - Enumerating ChildObjects 80
  - Implicit Properties 78
  - Problems with Ordinal Identifiers 84
  - Using a VBScript DP object repository 83
  - Using description objects 78
  - Using Description Strings 80
  - When to use Descriptive Programming 78
  - Working with Multiple Browser 86
- Desktop.CaptureBitmap 164
- DisableReplayUsingAlgorithm 342
- Document Object 197
- DotNetFactory 377
  - .Net Arrays Revisited 387
  - Accessing the Registry 386
  - Converting Images to other File Formats 390
  - Evaluating Keyboard Control Key Status 386
  - Getting Computer Information 385
  - Getting user Input using .NET Forms 392
  - Passing Enums 383
  - Passing Parameters to .NET Objects 382
  - Passing Parameters to Class Constructors 381
  - Ping an IP Address 386
  - Playing a Wav File 385
  - Sending Emails using .NET 389
  - Using the DotNetFactory Object 381
  - Working with .NET Arrays 384
  - Working with .Net Date and Time Formatting 388
  - Working with .Net Queues 388
  - Working with .Net Stacks 387
  - Working with the Clipboard 385
- DownloadAttachments 329
- Dynamic Arrays 142
- E**
  - Encrypt 55
  - EnumerateAllTestSets 334
  - EnumerateAllTestsinTestPlan 333
  - EnumerateSettings 341
  - Environment variables 45
    - BROWSER\_ENV 49
    - ExternalFileName 46
    - LoadFromFile 46
    - Types of Environment variables 45
    - URL\_ENV 49
    - Value 46
  - Error Handling 107
  - Eval 144
  - Excel.Application 136, 181, 182, 254, 258, 347, 353
  - Excel Automation Object Model 253
  - Execute 51, 79, 143
  - ExecuteFile 70
  - ExecuteFileGlobal 71
  - ExecuteGlobal 72
  - ExecuteGlobalCode 72
  - Executing a Stored DB procedure 249
  - Executing code when Script ends 346
  - Exist 153
  - ExitAction 39
  - Exporting XML to a DataTable 237
  - Extend 356
  - extendWebTable 356
  - Extern.Declare 171
  - ExternalDataTablePath 342
  - Extern object 171
- F**
  - Filter 131
  - FilterStr 139
  - FindAttachment 328
  - FindWindow 184
  - FindWindowEx 184
  - FireEvent 154
  - FirstGlobalIteration 342
  - Fixed length arrays 142
- G**
  - GenericSet 60
  - GetAddressesFromMail 279
  - GetBusinessDays 134
  - GetColumnName 265



GetCurrentTestPath 332  
 GetCurrentTestSetPath 335  
 GetCursorPos 179  
 GetDC 173  
 GetDOMObjectByPath 205  
 getElementsById 198  
 getElementsByName 198  
 getElementsByTagName 198  
 GetEnvironmentVariable 172  
 GetFileName 140  
 GetFilePath 140  
 GetForegroundWindow 171, 172  
 GetIECOMByhWnd 299  
 GetIECOMByURL 299  
 GetIEObjectFromhWnd 262  
 GetObject 136  
 GetParameter 34  
 GetParentOfElement 218  
 GetPixel 173  
 GetRandomNumberByLen 300  
 GetROProperty 8, 17  
 GetRowCount 35  
 GetSheet 33  
 GetSMTPAddress 280  
 GetStrBetweenChar 137  
 GetStrFromChar 137  
 GetSubFolderByPath 271  
 GetTOProperties 16  
 GetTOProperty 17  
 GetVarXML 48  
 GetWindowDC 173  
 GetWordFromPos 138  
 GlobalIterationMode 342  
 GlobalSheet 32

## H

Hex 132  
 Hour 132  
 HowManyWeekDays 133  
 HTML Button 199  
 HTML Checkbox 200  
 HTML ComboBox 199  
 HTML Element Collection 197  
 HTML Element Object 197  
 HTML Link 199  
 HTML List Box 199  
 HTML Node 196  
 HTML Radio Button 200

HTML Table 201  
 HTML Text Box 199

## I

IEDownloadFile 312  
 InStr 124  
 InStrRev 126  
 Internet Explorer 295
 

- EnumerateIE 298
- Popup Blocker 305
- Using IE to get user input 302

 InternetExplorer.Application 296, 297, 301  
 IsConnected 322  
 IsEnvExist 47  
 isParameterExists 33  
 IsRegEqual 117

## J

Join 128  
 JScript 357
 

- Passing variable number of arguments to a function 358
- Scope differences 358
- Try...Catch block 359
- Working with JS classes 360
- Working with JScript Arrays 360

## K

keybd\_event 173  
 KeyDown 174  
 KeyPress 174  
 KeyUp 174

## L

LastGlobalIteration 342  
 launchreport 343  
 Lcase 123  
 Left 124  
 Len 124  
 Library files 69
 

- Associating a Library globally with a Test 69
- Dynamically Loading libraries locally at run-time 70, 73

 LoadFromFile 52  
 LocalSheet 33  
 LTrim 127  
 LTrimW 120

## M

MakeObjVisible 348

MapVirtualKey 173  
 MercuryTimers 375  
 micclass 80  
 micGreaterThan 375  
 micGreaterThanOrEqual 375  
 micLessThan 375  
 micLessThanOrEqual 375  
 micNotEqual 375  
 micRegExpMatch 375  
 Mid 124  
 Minute 132  
 Modal dialog boxes 184  
 Month 132

## N

NewSync 311  
 Now 132

## O

Object Identification 21, 77  
 Object Spy 19, 367  
 ObjRepType 342  
 Oct 132  
 On Error Resume Next 107  
 OnReplayError 342  
 Open Test Architecture (OTA) 324  
 Optional arguments in VBScript 144
 

- Using an Array of (Key, Value) pairs 146
- Using an Array of Key:=Value pairs 147
- Using an Array of values 145
- Using Null/Empty values 144

 OptionalStep Object 56  
 Outlook.Application 270  
 Outlook Object model 269  
 Outlook Security Dialogs 276

## P

PathFinder.Locate 57, 330  
 PathFinder Object 56  
 PostMessage 184  
 Print Statement 372
 

- Clearing the Print Window 373
- Hiding the Print Window 372
- Programmatically Capturing the Print Log text 374
- Showing the Print window 373

## Q

QCUtil 370

QCUtil Object 322  
 Quality Center 319
 

- ResolveRelativePath 321

 QuickTest.Application 168, 189, 192, 325, 337, 338, 345

## R

RandomNumber Object 57  
 Recovery\_LoginProxy 103  
 Recovery object 105  
 RegExp 117  
 RegisterUserFunc Object 60  
 RegularExpression 17  
 Regular Expressions 111  
 ReleaseDC 173  
 Replace 127  
 ReplaceMultipleByOne 141  
 ReplayType 156  
 Reporter.Filter 160  
 Reporter.LogEvent 166  
 Reporter.ReportEvent 161  
 Reporter.ReportPath 164  
 RepositoriesCollection 377  
 rfDisableAll 160  
 rfEnableAll 160  
 rfEnableErrorsAndWarnings 160  
 rfEnableErrorsOnly 160  
 Right 124  
 RTrim 127  
 RTrimW 119  
 RunAction 39

## S

Scripting.Dictionary 71, 148, 149, 166, 169, 257, 298, 349, 354  
 Scripting.FileSystemObject 48, 161, 167, 168, 169, 240, 255, 329, 345, 347, 405  
 Seconds 132  
 SelectRadioByTextIndex 206  
 SendMessage 172  
 SetActiveWindow 184  
 SetCurrentRow 34, 35  
 SetForegroundWindow 173  
 SetSecure 56  
 Setting Object 57  
 SetTOProperty 17  
 ShowWindow 176  
 SnapshotOnErrorActiveUI 342  
 SnapshotReportMode 163

Space 128  
 Split 128  
 SplitEx 129  
 StopAndReRunQTP 344  
 StrComp 123  
 String 128  
 StrReverse 124  
 Sync 151  
 Synchronization Points 151  
 SystemUtil 370  
     BlockInput 370  
     UnBlockInput 370  
 SystemUtil.Run 86  
 SystemUtil Object 59

## T

TDAPIOle80.TDConnection 326  
 TDConnection 322, 324  
 Test Arguments 191  
 Test Results 159  
     Accessing Test Results at the end 167  
     Configuring QTP to generate HTML results 164  
     Converting XML Results to HTML 164  
     Inserting Files in Test Results 161  
     Inserting Snapshots in Test Results 162  
 Time 132  
 Timer 134  
 Trim 127  
 TrimAll 138  
 TrimW 120  
 TSRFilePath 343  
 TypeName 135

## U

UBound 143  
 Ucase 123  
 URLDownloadToFile 176  
 UseExternalDataTablePath 342  
 Useful Tools 409  
     IE WebDeveloper 413  
     Internet Explorer Session Cookies 413  
     Notepad++ 410  
     QTP Script Editor 409  
     QTP Uninstaller 411  
     RAD Regular Expression Designer 416  
     ScreenCapture API 412  
     Test Design Studio (TDS) 415  
     VB 2 QTP API Converter 413  
     VBSEdit 410

User-defined Objects 22

## V

ValueByRow 34  
 VarType 135

## W

Wait 152  
 WaitProperty 152  
 WebRegSelectValue 118  
 Web Tables 211  
     Access using OR 216  
     Asynchronous Table loading 223  
     Clicking inside a WebTable 221  
     Exporting WebTable to a DataTable 223  
     Finding a cell in a Table 225  
     FindTextBySearch 226  
     GetTableFromElement 219  
     Getting a QTP Table from a element 224  
     Getting the location of an object inside the table 219  
     GetWebTableFromElement 224  
     Using an object inside the table 218  
     Using Index 213  
     Using innerText/outerText 213  
     Using Name or HTML ID 213  
 WebUtil Object 59  
 Weekday 133  
 WScript.Shell 168, 183, 187, 282, 295, 305, 309, 345, 405, 406

## X

XMLAttribute 232  
 XML Attributes 232  
 XMLAttributesColl 232  
 XML Child nodes 232  
 XMLData 232  
 XMLElement 232  
 XMLElementsColl 232  
 XMLItemColl 232  
 XML Node values 232  
 XML Root node 231  
 XMLUtil 232  
 XMLUtil.CreateXML 232  
 XMLUtil.CreateXMLfromFile 233

## Y

Year 132

## 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>

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

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