TEST AUTOMATION USING UNIFIED FUNCTIONAL TESTING

Learn automation on web based application

Automation related interview questions

Step by step instructions

Real life examples

Explore latest version of

QTP

NAVNEESH GARG

Test Automation using HP UnifiedFunctional Testing (UFT)11.5

Navneesh Garg

- Step by step guide
- Learn automation on a web based application
- Real life examples
- Step by step instructions
- Automation related interview questions

Explore latest version of HP QTP

ISBN - 978-0-9922935-1-2

Publisher: AdactIn Group Pty Ltd.

Contents

About the Author

Navneesh Garg

Navneesh Garg is a recognized test automation architect and corporate trainer, specializing in test automation, performance testing, security testing, and test management. Certified in HP QTP, HP Quality Center, HP LoadRunner, and IBM Rational Functional Tester and as a Certified Ethical Hacker, he has proven success in managing a wide array of testing and automation projects, leading test teams, designing automation frameworks, and providing technical training. As a tool specialist, he has worked on a variety of functional automation tools including Selenium, TestComplete, TestPartner, SilkTest, Watir, RFT, HP QTP/UFT and on varied technologies including Web, Java, Dot-net, SAP, Peoplesoft, and Seibel.

He is an entrepreneur and founder of several successful IT companies, which encompass the AdactIn Group, CresTech Software, and Planios Technologies.

As an experienced corporate trainer, he has trained professionals in HP QTP/UFT and other test tools across wide range of global clients like Macquarie Bank, Corporate Express, Max New York Life, Accenture, NSW Road and Maritime Services, Australian Dept of Education, HCL Technologies, Sapient, Fidelity group, Adobe Systems, and many more. He has training experience in diverse geographies like; Australia, India, Hong Kong, and USA.

He has worked with Mercury (now HP) and led the design of instructor led courses for QuickTest Professional tool and HP Quality Center, as a subject matter expert. He holds Bachelors in Computer Science from NSIT, Delhi University, India.

As a technical test delivery head for his company, he has led and managed functional automation testing and performance testing teams across a wide range of domains, using commercial tools and open source tools. In addition he has also designed several advanced automation frameworks using these tools.

Preface

My motivation for writing this book stems from my hands on experience in the IT and testing industry and the experience I have gained as an automation consultant working in numerous complex automation projects. It disheartens me when I come across automation and frameworks being implemented in a very novice fashion with basic automation principles and guidelines not adhered to. In such cases, rather than automation providing a true advantage to these practitioners, it leads to major maintenance effort and automation failures. Of course it leads to a loss of faith too in test automation as a test activity.

The key objective of this book is to get the audience to understand, the **in depth internals of automation,** without getting into a lot of theory.

Scope of Topics

HP recently launched the new version 11.5 of QuickTest Professional (QTP). In this new version, traditional QTP has been combined with HP Service Test (which was a separate tool to test web services and API). Hence, this new version is called HP UFT (Unified Functional Testing). UFT has all the same features, which QTP 11.0 had and more. We will be using the word UFT or QTP interchangeably through the book. **HP Service Test features are out of scope** of this book.

As part of scope of this book we will primarily discuss **GUI based Tests,** which are similar to QuickTest Tests in previous versions of QTP. API tests, Business Process Test, and Business Process Flow, are not in the scope of the book.

Any references made to HP QTP in this book, refer to features of QuickTest Professional version 11.0

My intent here in this book is to teach all the key features of UFT, and cover all crucial aspects of the same, which are used to **create effective automation frameworks**.

Key Audience

The target audience for this book are functional testers and beginner test engineers, who want to **learn UFT quickly** and in the right way, which will help them create an effective automation framework.

Unique Selling Points of the Book

Test Automation using HP Unified Functional Testing (UFT), is the **first book released globally** on HP UFT, which is the latest version of the HP test automation tool, Quick Test Professional.

This book has been designed with the objectives of **simplicity and ease of understanding**.

Another major highlight of this book is that you will be using our **custom developed web based application**. As a corporate trainer, every time I used to go and deliver QTP/UFT training with the standard desktop based flight reservation application (that comes by default with HP QTP/UFT), participants wanted to use something, which is different, preferably a web-based application, which is what most of them would eventually work on in the real world. They wanted an application, which had more features and test scenarios that could be automated. As part of this book, we have developed a web-based application, which has much more enhanced test scenarios, and will bring you very close to real-time automation.

This book follows a **unique training based approach** instead of a **regular text book approach**. As a step by step guide, it guides the student through every step of the exercises with the help of snapshots.

Another differentiator is that I have tried to include **lot of practical examples and issues,** which I and most of automation testers see in day to day automation. These experiences will give you an insight into what challenges you could face when you start doing real automation. Practical examples in this book cover how to use most of the features within HP UFT.

The book also covers the most **common interview questions,** asked at test automation interviews.

It also covers aspects of **Integration of HP UFT with HP ALM** (Quality Center) platform. It explains how we can connect from UFT to HP ALM, store our scripts in ALM, execute our scripts from ALM, and view test results.

Sample Application and source used in Book

The Sample application used in the book can be accessed at the below URL

www.adactin.com/HotelApp/

Source code used in the book can be found at below link

www.adactin.com/store/

Feedback and Queries

For any feedback or queries you can contact author at www.adactin.com/contact.html or email navneesh.garg@adactin.com

Order this book

For bulk orders, contact us at **orders@adactin.com**
You can also place your order online at **adactin.com/store/**

Acknowledgements

I would like to thank my family (my parents, my wife Sapna, my wonderful kids Shaurya and Adaa) for the continued support given to me, without whom this book would not have been possible.

Special thanks to S. Constâncio Colaço and Philip Smith for their reviews and feedback, which immensely helped as I worked on this book.

I would also like to thank my colleagues and clients for the inspiration, knowledge, and learning opportunities provided.

1 Introduction to Automation

Introduction

In this chapter we will talk about automation fundamentals and understand what automation is and why we need automation. Another important objective of this chapter is to understand the economics of automation and determine when we should do automation in our projects. We will also get an insight on a few popular commercial and open source automation tools available in the market.

Key objectives:

- What is automation?
- Why automate? What are the benefits of automation?
- Economics of automation.
- Commercial and Open Source automation Tools.

1.1 What is Functional Automation?

Automation testing is to automate the execution of manually designed test cases, without any human intervention.

The purpose of automated testing is to make manually executed functional testing, cost effective and quicker. Most often, we rerun tests that have been executed before to test functional correctness of the application, also called regression testing. Think of a scenario where you need to validate the username and password for an application, which has more than 10,000 users. It can be a tedious and monotonous task for a tester. This is where the real benefits of automation lie. We want to free up the time of a manual functional tester to perform other key tasks and still provide extensive coverage to the overall test effort.

When we use the term "automation", I see people get confused, as they refer to two varied testing types, functional and performance. Automation covers both

- Functional Automation – Used for automation of functional test cases in the regression test bed.

- Performance Automation – Used for automation of non-functional performance test cases. What this usually means is measuring the response time of the application under considerable (say 100 users) load.

Functional automation and performance automation are two distinct terms and their automation internals work using different concepts, and hence different tools are used.

For the scope of this book, we will be only concentrating on functional automation. Whenever we refer or use automation in this book, we actually mean functional automation.

1.2 Why do we Automate?

Find below the key benefits of Functional Automation

1. **Effective Smoke (or Build Verification) Testing**

Whenever a new software build or release is received, a test (generally referred to as "smoke test" or "shakedown test") is run to verify if the previously working functionality is still working. It sometimes can require numerous hours to complete an entire smoke test, only to realise that a faulty software build has been received. This has now resulted in wasted testing time, because the build has to be rejected, and testing has to start all over again.

If the smoke test is automated, the smoke test scripts could be run by the developers to verify the build quality, before it is handed over to the testing team, saving valuable testing time and cost.

2. **Standalone - Lights Out Testing**

Automated testing tools can be programmed to kick off a script at a specific time.

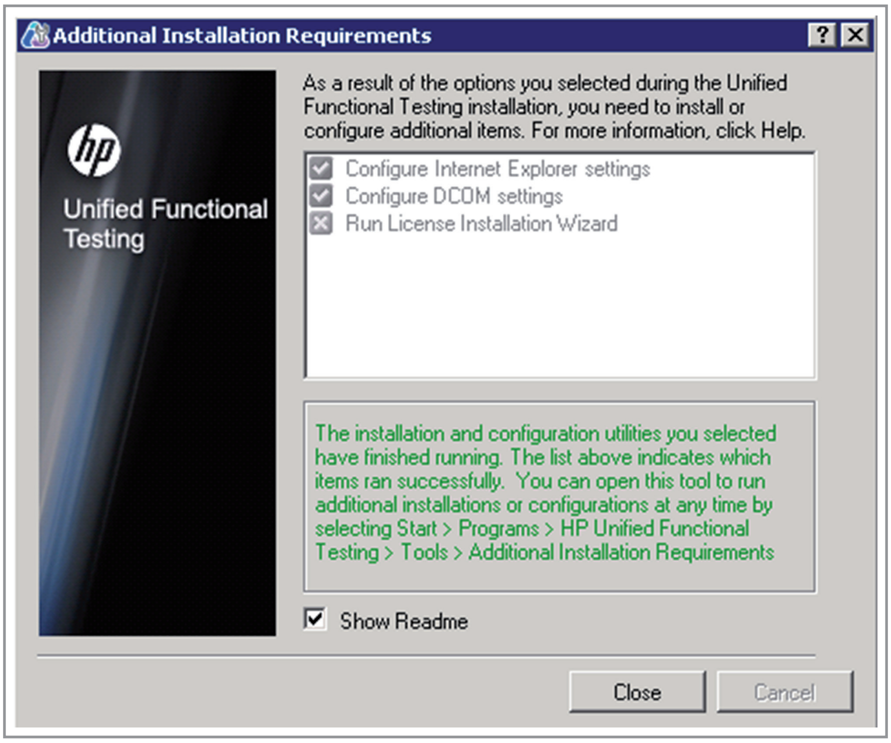
Consider the fact that automated testing can be run standalone and be kicked off automatically if needed, overnight, and the testers can analyse the results of the automated test, the next day they are in the office.

3. **Increased Repeatability**

Often a test is executed manually that uncovers a defect, only to find out that the test cannot be reproduced, i.e. the tester forgot which combinations of test steps led to the error message and is not able to recreate the defect. Automated testing scripts take the guess work out of test repeatability.

4. **Testers can Focus on Advanced Issues**

As tests are automated, automated scripts can be base-lined and rerun for regression testing purposes, which generally yields less new defects, than testing over newly developed features. Testers can focus on newer or more advanced areas, where the most defects can be uncovered, while the automated testing script verifies the regression testing area. New features are incrementally added to the automated test regression test suite.


Additional Installation Requirements

As a result of the options you selected during the Unified Functional Testing installation, you need to install or configure additional items. For more information, click Help.

- ✓ Configure Internet Explorer settings
- ✓ Configure DCOM settings
- ☒ Run License Installation Wizard

The installation and configuration utilities you selected have finished running. The list above indicates which items ran successfully. You can open this tool to run additional installations or configurations at any time by selecting Start > Programs > HP Unified Functional Testing > Tools > Additional Installation Requirements

☑ Show Readme

Figure 2-8– HP UFT Installation – Close Additional Installation Requirements

18. You can read the Readme file, which opens automatically.
19. You can now access HP UFT from **Start** -> **All Programs** → **HP Software** → **HP Unified Functional Testing** → **HP Unified Functional Testing**
20. If you are using the trial version of the software, you will get License Expiration warning dialog, once you launch HP UFT. Click on **Continue** to keep using the tool for the trial duration.

Note: If you do not see Continue button in the license expiration dialog (instead you see Cancel button) , it is possible that you might have a previous version of HP QuickTest installed on the machine, because of which you would not be able to use the trial software. There is no workaround. In that case you cannot use trial version of the software on that machine, and you would need to purchase the software.

Note: Refer to HP UFT Installation guide, in case you need more detailed steps or face any issues with installation.

2.2 Sample Application Walkthrough

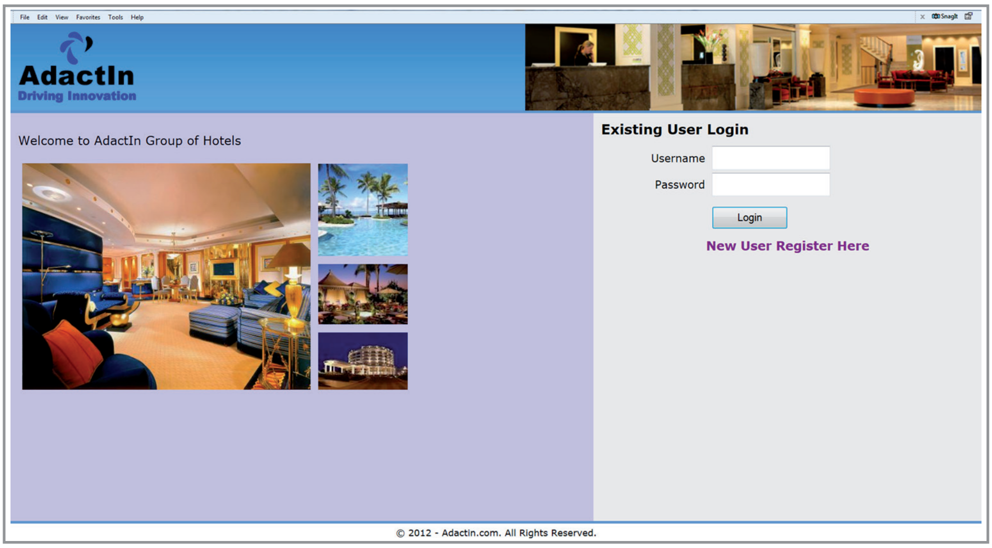
As part of this book, we will be working through a web based sample application. The reason why we planned to use our custom built web based application, was that 80-90% of applications tested and automated are web based applications. So we will have a much closer and a better understanding of how we need to automate web-based application.

Our sample application is a simple hotel booking web application, which has the following key features

- Search for Hotel
- Book a Hotel
- View Itinerary
- Cancel Booking

Let us browse through the application

1. Launch IE and enter URL www.adactin.com/HotelApp to see Login page.



Welcome to AdactIn Group of Hotels

Existing User Login

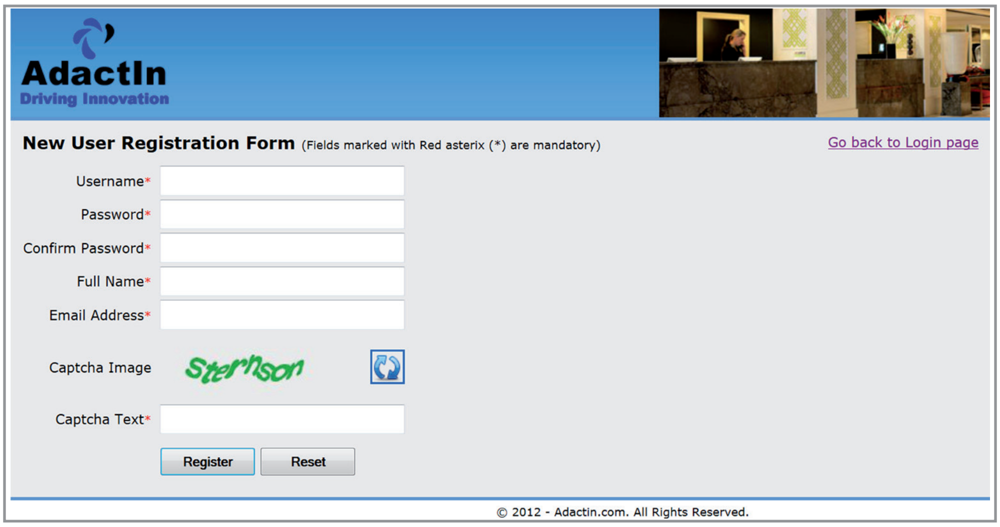
Username

Password

© 2012 - Adactin.com. All Rights Reserved.

Figure 2-9 – Application Login Page

2. Click on "New User Register Here" to go to Registration page.



New User Registration Form (Fields marked with Red asterix (*) are mandatory)

Username*

Password*

Confirm Password*

Full Name*

Email Address*

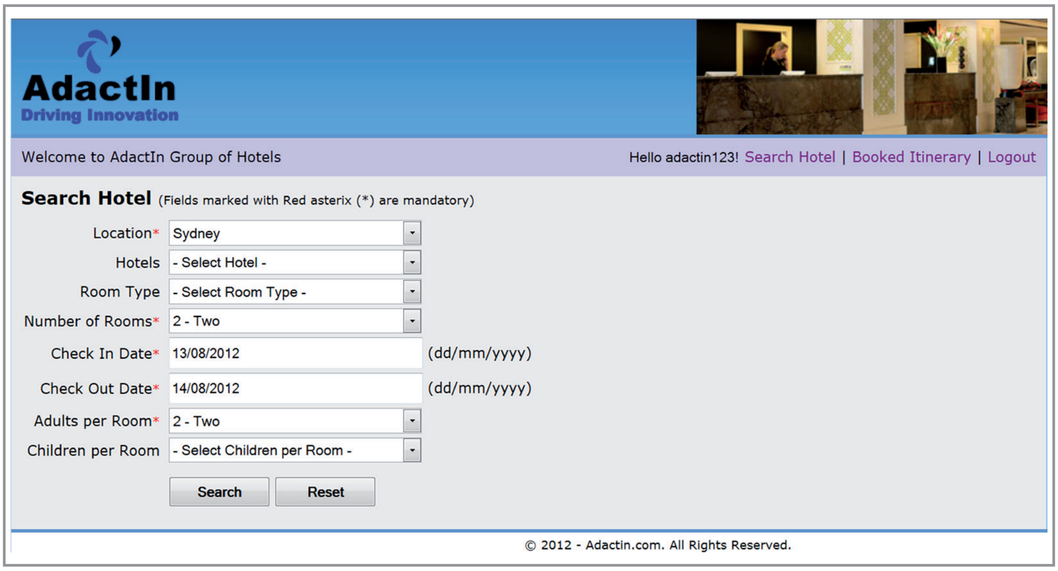
Captcha Image

Captcha Text*

© 2012 - Adactin.com. All Rights Reserved.

Figure 2-10 – Application Registration Page

3. Register yourself by entering all the fields. Remember the username and password as you will be using this username/password to login to application and remaining part of automation.
4. After you register, an automatic email will be sent to your email-id for confirmation. Incase you do not receive the email, re-verify it in junk folder as email might have gone to your junk folder.
5. Click on the confirmation link in email
6. Go to Login page link.
7. On the Login page use the username/password with which you have registered earlier, and click on the Login page. You will come to Search Hotel Page.
8. Search for Hotel-
   i. Select a location e.g. Sydney
   ii. Select room nos. e.g. 2
   iii. Select adult rooms e.g.2
   iv. Click on Search button



Welcome to AdactIn Group of Hotels

Hello adactin123!

Search Hotel (Fields marked with Red asterix (*) are mandatory)

Location*	Sydney
Hotels	- Select Hotel -
Room Type	- Select Room Type -
Number of Rooms*	2 - Two
Check In Date*	13/08/2012 (dd/mm/yyyy)
Check Out Date*	14/08/2012 (dd/mm/yyyy)
Adults per Room*	2 - Two
Children per Room	- Select Children per Room -

© 2012 - Adactin.com. All Rights Reserved.

Figure 2-11 – Application Search Hotel Page

9. Select a Hotel-
   i. Select one of the Hotel Radio Button, e.g. select radio button next to Hotel Cornice

4.3 Recording Script

Alright! Let us record our first script now.

Note: Make sure web-ins are loaded and selected, in add-ins dialog when you launch UFT.

Note: Make sure all other applications and browsers except UFT are closed. This is not mandatory, but at a beginner level it helps you to make sure you do not incorrectly record.

1. Before we start recording, we will launch IE and enter URL. So launch IE and launch URL www.adactin.com/HotelApp in your browser page to see the websites login page.

Note: For our future exercises, we will assume application login page is visible and that the browser is launched.

2. Click on **Record** Button or Go to **Record** → **Record** and click.

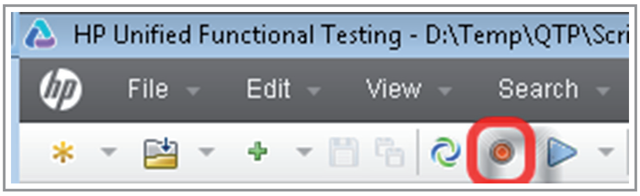
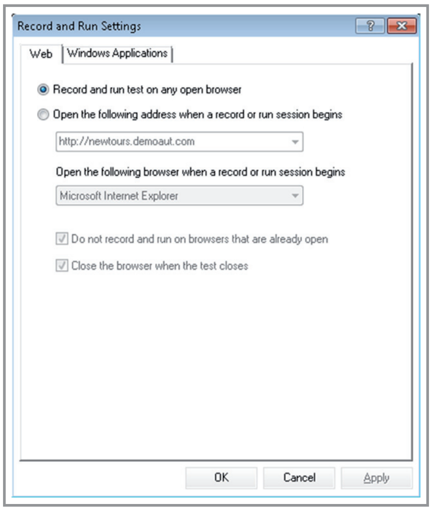


Figure 4-7 – Recording Icon

3. In the **Record and Run settings** dialog, select Web Tab and select **Record and run test on any open browser**.



Record and Run Settings

Web | Windows Applications

[✓ Record and run test on any open browser]

Open the following address when a record or run session begins

http://newtours.demoaut.com

Open the following browser when a record or run session begins

Microsoft Internet Explorer

[✓] Do not record and run on browsers that are already open

[✓] Close the browser when the test closes

OK | Cancel | Apply

Figure 4-8 – Record and Run Setting Dialog

Record and run test on any open browser: This option enables you to record and run on any browsers which are open. This is the most common and widely used option.

Open the following address when a record or run session begins: You can define specific URL that you would like to open when your script records or runs. This is not a frequently used option.

4. Click **OK**.

Now you are in recording mode. Anything you do from now on will be recorded as an action in UFT script. You would see a red "Recording GUITest1" flashing dialog comes, giving an indication you are in recording mode.

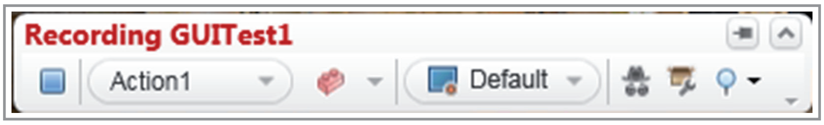


Figure 4-9 – Record Toolbar

5. Assuming that application is already open in IE browser with login page visible, and perform below steps.
   a. Login (Use the username/password with which you have registered earlier).
   b. Search for Hotel.
      i. Select a location e.g. Sydney
      ii. Select room no: e.g. 2
      iii. Select adult rooms e.g.2
      iv. Click on Search button
   c. Select a Hotel.
      i. Select one of the Hotel Radio button e.g. select radio button next to Hotel Creek
   d. Book a Hotel.
      i. Enter First Name
      ii. Enter Last Name
      iii. Enter Address
      iv. Enter 16 digit Credit card no:
      v. Enter Credit card type
      vi. Enter Expiry Month
      vii. Enter ExpiryYear
      viii. Enter CVV number
      ix. Click on Book now

6

Parameterization – Data Driven Test

Introduction

Many a times when we are performing regression testing, we need to repeat the same test cases with different types of data. Now this can be a monotonous and time consuming task, depending on how many different data sets are required with which we need to verify that data using the tests.

Example

Let us take an example:

I worked for one of our retail domain client, which had more than 2000 stores in the country, as part of testing and automation team. They developed a point of sales system and once this was manually tested, they gave us a list of more than 10000 username and password pairs, and asked us to set them up in the system. As a testing team, our task was to verify if all usernames and passwords are setup correctly. We were given a target of .01% failure threshold. If we have to verify all this manually, assuming we would verify 1 username/password combination every 1 minute (as there were couple of validations we had to do once logged in), it would have taken us 5000 minutes or approximately 20-24 days of man effort, get the idea of how laborious and time consuming that task would be?

Solution: Wouldn't it be great if you have a script using which I could pick up the first username and password entered from an excel sheet, log the user in and perform all the validations and all of this without manual intervention?

An even better solution, if we could create a script to iterate across all the 10000 usernames and passwords, wouldn't that be a better solution. This is exactly what did, we created an automation script using UFT, which took us less than 4 hours to develop , ran it overnight, got the failed records, again re-tested the records once fixed and delivered it to the customer with 0% issues. The concept of running the same script with multiple dataset values is called Parameterization.

Any test case, which needs to be executed multiple times with different data values are ideal candidates for automation.

Key objectives of this chapter are:

- How to Parameterize my Script.
- Script Execution and Result Analysis.

6.1 How to Parameterize a UFT Script

Scenario: As a test workflow, we need to book a Hotel for all the possible available cities, in our case, Sydney, Melbourne, Brisbane, and Adelaide. So we want our script should be able to iterate once each for the above cities.

Should we create multiple scripts for every city?

The answer is No! We should use the same script and get it to be executed for multiple sets of data.

Let us see how to solve the above problem using Parameterization:

1. Open one of previously created scripts **MyFirstTest** which was created in your last exercise
2. Go to **File > Save As** and save the script as "**Parameterization**"

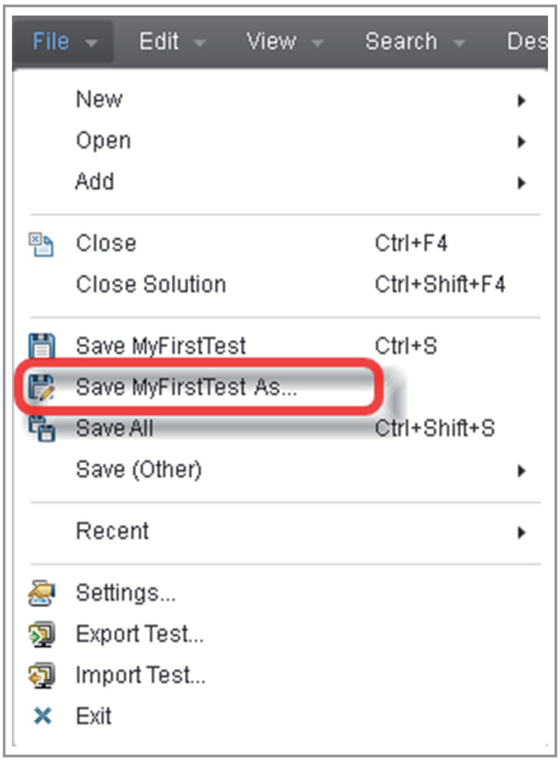


Figure 6-1– Save Test Option

3. Make sure you are in Keyword view of UFT (Select **View** → **Keyword View**).
4. Locate the line with Item **location** in the Keyword view.
5. Go to column **value** next to location line and click in the value column, and select the city.

10

Working with Shared Repository

Introduction

We now know how UFT works as a tool, and recognizes objects based on object properties. But when we perform recording, objects get added to local object repository. For instance when we login, username, and password objects get added to our local object repository.

Now let us think, if we have more than 200 UFT scripts automated, these objects would be duplicated in each of these scripts. This means we are redundantly duplicating these objects in our scripts.

The problem does not end here. What if for due to some business reason, the developer changes the property of these objects? This would result in all our 200 scripts failing. In order to fix this issue we will need to go to each and every script, and change its local repository. This can be a nightmare to maintain!

A better solution would be if we kept all the required objects in an external shared location and all the scripts could just use these shared location objects. This will certainly avoid redundancy. Also, if any object changes, we would only need to change the object once and the script will be up and running again.

Example

At one of our clients, where we had to implement automation, we were given 100 existing automation scripts created using QTP/UFT. We were told they used to work 3 months back, but now, after they were run on new builds, they fail. We were asked to fix the scripts. Guess what we found? All the scripts were using local object repository. When we identified the objects that were causing the script to fail, we discovered that the same object was used in all 100 scripts. So the object had to be modified at least 100 times as it was being used in all the scripts. But there were at least 100 objects, which had changed. Adding to this, they informed us that UI changes were still happening and that the objects will change again. Our recommendation to them was to hold-on and to re-do the scripts using shared object repository, since the same effort that we would invest now would be required again when we get a new application build, with updated objects. Yes, it did mean that most of the previous effort already done had been wasted, but our re-scripting approach using a shared object repository approach made sure scripts maintenance worthy for future.

In this chapter we will discuss how to create and add objects to shared object repository, and how to maintain the shared object repository.

Key Objectives

- Create and add objects to Shared Object Repository
- Rename logical names of objects
- Associate Shared Repository to script
- Recording of script using Shared Repository

10.1 Create and Add Objects to Shared Object Repository

In this section we will see how to create, and add objects to Shared Object Repository.

1. Go to **Resources → Object Repository Manager**.

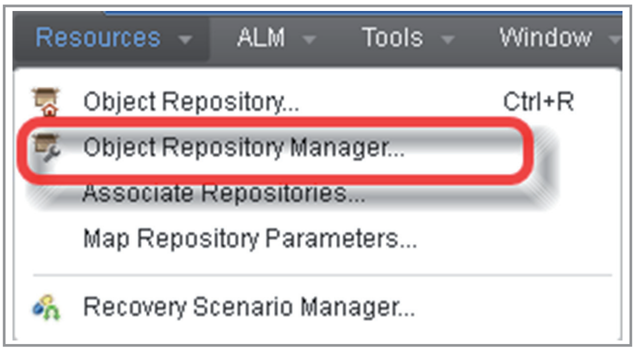
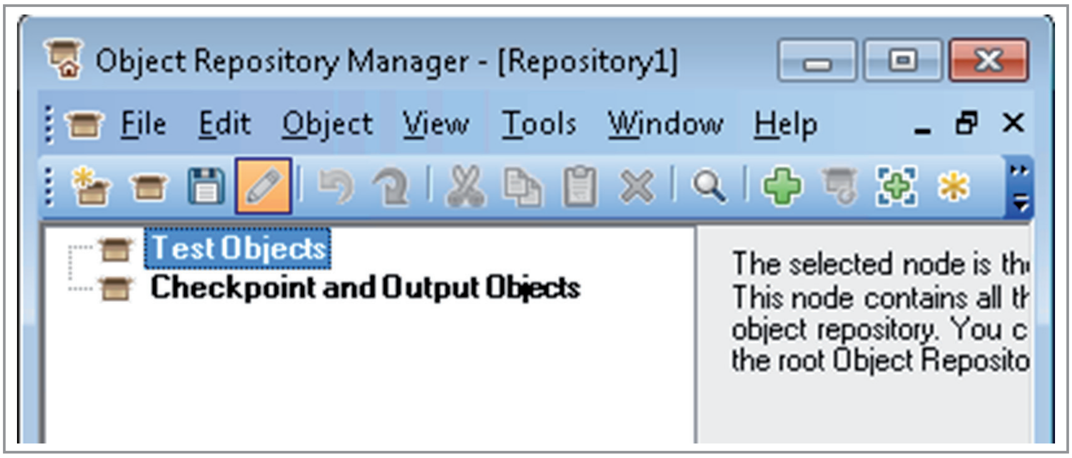

Resources
ALM
Tools
Window
Object Repository...
Ctrl+R
Object Repository Manager...
Associate Repositories...
Map Repository Parameters...
Recovery Scenario Manager...

Figure 10-1 – Object Repository Manager Menu

2. You will see Object Repository Manager Dialog Opens up


Object Repository Manager - [Repository1]
File
Edit
Object
View
Tools
Window
Help
Test Objects
Checkpoint and Output Objects

The selected node is th
This node contains all th
object repository. You c
the root Object Reposito

Figure 10-2 – Object Repository Manager (ORM) Dialog

Note: We will create/modify shared object repository using Object Repository manager.

12

VBScript Fundamentals

VBScript is the scripting language used in UFT editor view. We can write any VBScript statement in UFT editor view and it will be executed. This makes UFT very powerful. VBScript being a global language, we can find plenty of sample code on Google and use it in our UFT GUI script.

Example

Recently we automated some scenarios where we had to create an automation test report in a MS Word document format with all snapshots, which we had to forward to the development and business team. We used VBScript and MS Word object model to easily accomplish this.

Using VBScript we can interact with external third party applications (Word, Excel, PDF, Text file, Database), and write custom code in our script, which make our script more intelligent and worthwhile.

Having said this, VBScript is still a very simple language. We do not use any complex pointers, data structures, exception handing blocks or object instantiation constructs. Many testers fear using complicated programming principles. This makes UFT very easy to use, by not only programming experts but also by functional testers or business analysts.

Key Objectives:

- VBScript Basics and data types
- VBScript Conditional Statement, Loop Statements, Procedures
- Commonly used VBScript string functions
- Sample VBScript programs

12.1 VBScript Basics and DataTypes

What is VBScript?

- VBScript is a scripting language
- A scripting language is a lightweight programming language
- It is primarily used for client side scripting within html pages
- VBScript is a light version of Microsoft's programming language, Visual Basic

VB is different from VBScript. VB is a programming language with a full blown programming environment like integrated development environment and VBScript is a simpler scripting language, which is normally written in a text editor like notepad.

HP UFT acts as an editor for VBScript. Any VBScript code can be written in HP UFT editor view and executed. Using VBScript, we can read/write from excel sheets, text files, word documents, databases, and other external sources. It's important to understand that these are accomplished by the VBScript language and not by UFT. We can however find such read/write code snippets by doing a Google search and just copy pasting them in UFT editor view to execute the scripts.

Windows Script

VBScript talks to host applications using windows script. With windows script, browsers, and other host applications do not require special integration code for each scripting component. Windows script enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer. Microsoft provides run-time support for VBScript.

Data Types:

What is a Variable?

A variable is a name given to a location in memory, where some data used by a program is stored. You can refer to a variable by name to see or to change its value.

Variables can hold different types of data: numbers, characters, strings, and some special complex data. These categories are called data types.

VBScript keeps it simple. In VBScript there is only one datatype "**Variant**". In this variant type, you can store numbers, characters, strings, boolean and other data types. Surely this makes things simpler!

For example: The books detail of a library, stores the id of the book, and name of the book and for storing these data sets it uses variables.

book_id
book_name

VBScript is a loosely typed language, which means you do not have to declare the variables before using them in your scripts.

Rules for Variable Names:

- Must begin with a letter
- Cannot contain a period (.)
- Cannot exceed 255 characters

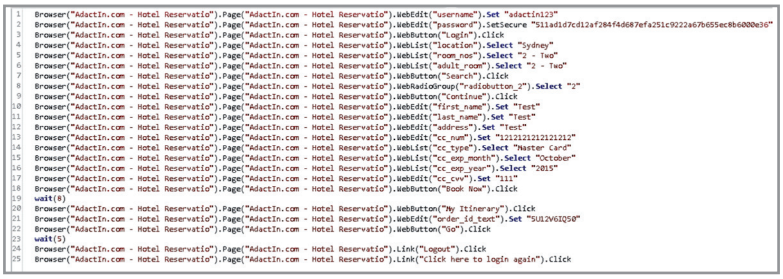
14.3 Using GetROProperty to Capture Dynamic Values

Capturing dynamic values using output value method, is an UFT feature to capture dynamic value using the UFT user interface. It is more relevant for automation testers who are not well versed with scripting.

GetROProperty method presents another alternate, but very popular method to capture dynamic values in the application. "RO" in GetROProperty method stands for "Runtime Object".

Let us look at how we can capture and resolve problems presented earlier in this chapter.

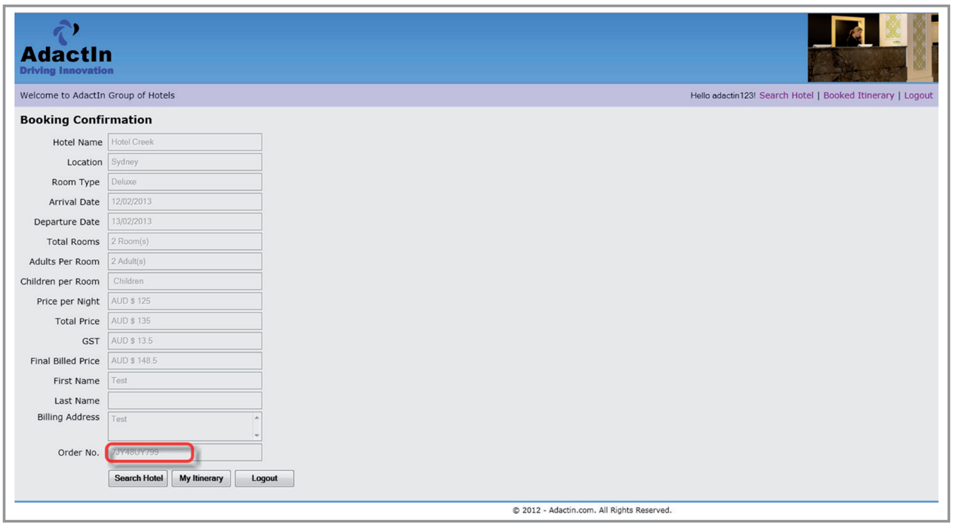
1. Open script "DynamicValueBasedscript" created in section 14.1 and do a **File → Save As** and save the script as "GetROpropertyscript".
2. Go to Editor View. (Go to **View → Editor**) You will notice the script below.



```
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("username").Set "adactin123"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("password").SetSecure "511ad1d7cd12af284f4d687efa251c9222a67b655ec8b6000e36"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Login").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("location").Select "Sydney"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("room_nos").Select "2 - Two"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("adult_room").Select "2 - Two"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Search").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebRadioGroup("radiobutton_2").Select "2"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Continue").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("first_name").Set "Test"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("last_name").Set "Test"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("address").Set "Test"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("cc_num").Set "1212121212121212"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("cc_type").Select "Master Card"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("cc_exp_month").Select "October"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebList("cc_exp_year").Select "2015"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("cc_cvv").Set "111"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Book Now").Click
wait(8)
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("My Itinerary").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("order_id_text").Set "5U12V6IQ50"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Go").Click
wait(5)
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").Link("Logout").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").Link("Click here to login again").Click
```

Figure 14-15– Editor View

3. Next step is to add Order No. field into Object Repository. Manually browse through the application so that you can see Order No. object.



AdactIn
Driving Innovation
Welcome to AdactIn Group of Hotels
Hello adactin123! Search Hotel | Booked Itinerary | Logout
Search Hotel
My Itinerary
Logout
© 2012 - Adactin.com. All Rights Reserved.

Booking Confirmation

Hotel Name	Hotel Creek
Location	Sydney
Room Type	Deluxe
Arrival Date	12/02/2013
Departure Date	13/02/2013
Total Rooms	2 Room(s)
Adults Per Room	2 Adult(s)
Children per Room	Children
Price per Night	AUD $ 125
Total Price	AUD $ 135
GST	AUD $ 13.5
Final Billed Price	AUD $ 148.5
First Name	Test
Last Name	
Billing Address	Test
Order No.	7JY48UY799

Figure 14-16– Order No object visible

4. Go to **Resources → Object Repository** and Click on **Add objects** button and Add Order Number object.

15

Reporter Object and Custom Checkpoint

In one of the previous chapters, we looked into how UFT provides standard checkpoint and others type of checkpoints, to verify functional correctness of the application. This is done at the basic level.

At an advanced level, we use custom checkpoints to verify the correctness of the application. What we mean by custom checkpoint is that using scripting we can manually compare our expected value with actual value and pass/fail the script. Now why do we need to do that? Reason is that it gives us more flexibility to manipulate the expected and actual values and compare values at will.

Example

We were once automating an accounting based application package. We had to test the reports feature of the application. For instance, if we had to verify the profit and loss report for a particular account, we had to first calculate the expected value from what was setup in various forms in the application, and fetch the actual value from reports and then perform comparisons. Now, we had to do this for lot of values we saw this in the report. Custom checkpoint made it lot earlier and flexible for us to perform these kinds of checks, because we could perform computation of expected values and also create loop statements to verify multiple values in the same script. This would have been hard using standard checkpoint as it did not allow runtime computations of expected values and we would have to add so many standard checkpoints.

Note: You can replace standard checkpoints and use just custom checkpoints.

In this chapter, we will see how to add a custom checkpoint. One of the key utility objects, which we'll use in this chapter, will be the Reporter Object. This object helps us to report pass, fail, or informational messages from within the script.

Key Objectives

- Reporter Object and ReportEvent Method Syntax
- Apply Custom Checkpoints on a Test Scenario

15.1 Reporter Object and ReportEvent Method Syntax

Reporter Object: The object used for sending information to the test results.

ReportEvent Method: It is the method of Reporter Object, which reports an event to the test results.

Syntax

Reporter.ReportEvent *EventStatus*, *ReportStepName*, *Details* [, *Reporter*

Argument	**Type**	**Description**
EventStatus	Number or pre-defined constant	Status of the report step: **0** or **micPass:** Causes the status of this step to be passed and sends the specified message to the report. **1** or **micFail:** Causes the status of this step to be failed and sends the specified message to the report. When this step runs, the test fails. **2** or **micDone:** Sends a message to the report without affecting the pass/fail status of the test. **3** or **micWarn**ing: Sends a warning message to the report, but does not cause the test to stop running, and does not affect the pass/fail status of the test.
ReportStepName	String	Name of the intended step in the report (object name).
Details	String	Description of the report event. The string will be displayed in the step details frame in the report.
ImageFilePath	String	**Optional**. Path and filename of the image to be displayed in the **Captured Data** pane of the Run Results window. Images in the following formats can be displayed: BMP, PNG, JPEG, and GIF.

Note: ImageFilePath is one of the latest features added in UFT, so that keep snapshot of result screens in your script

Example

The following examples use the **ReportEvent** method to report a failed step.

16.1 Working with text files

Usually we have seen automation frameworks will use text files to store configuration parameters.

To Read or Write to text files, you use the **File System Object** in UFT script.

Read from Text File

Let us see below steps or code you need to **Read from Text File.**

1. Create a **File System Object**

```
Dim fso
Set fso = CreateObject(“Scripting.FileSystemObject”)
```

The **FileSystemObject** (FSO) object model allows you to use the familiar *object.method* syntax with a rich set of properties, methods, and events to process folders and files.

2. Open a Text File using **OpenTextFile** method.

```
Dim fso, ts
var ForReading = 1;
Set fso = CreateObject(“Scripting. FileSystemObject”)
Set ts = fso.OpenTextFile(“D:\test.txt”, ForReading, True)
```

3. Use any of the below method to read from the text file.

- Read – Read a specified number of characters from a file.
- ReadLine – Read an entire line (up to, but not including, the newline character).
- ReadAll – Read the entire contents of a text file.

For example let us use **ReadLine**.

```
Dim fso, ts
var ForReading = 1;
Set fso = CreateObject(“Scripting. FileSystemObject”)
Set ts = fso.OpenTextFile(“D:\test.txt”, ForReading, True)
Do Until ts.AtEndOfStream
s = ts.ReadLine
msgbox s
Loop
```

4. Close the File and objects using **Close** method and setting objects to **Nothing**.

```
Dim fso, ts
```

5. Go to Editor View in UFT GUI Test script and we will use ImportSheet Method to Import Data from Excel Sheet.

ImportSheet Method

This method imports a sheet of a specified file, to a specified sheet in the run-time data table. Using this method we can import the data in our external excel file and get that data into the UFT scripts built-in data table.

The column headings in the sheet you import must match the data table parameter names in the action, for which the sheet is being imported. Otherwise, your test or component may fail.

Syntax

DataTab.ImportSheet(*FileName*, *SheetSource*, *SheetDest)*

Argument	Type	Description
FileName	String	The full path of the Excel table from which you want to import a sheet.
SheetSource	Variant	The name or index of the sheet in the file that you want to import. Index values begin with 1.
SheetDest	Variant	The name or index of the sheet in the Data Table that you want to replace with the *SheetSource*. Index values begin with 1.

Example

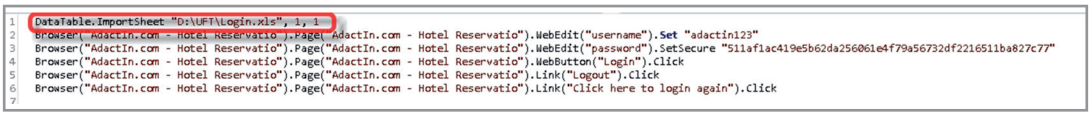
The following example uses the **ImportSheet** *method to import the first sheet of the name.xls* table to the name sheet in the test's run-time data table.

DataTable.ImportSheet "C:\name.xls" ,1 ,"name"

So let us go in the Editor view of our script and type below statement to import login data into our script.

DataTable.ImportSheet "D:\UFT\\Login.xls", 1, 1

Note: Please make sure to change the path based on, where you have stored the sheet. Your editor view code will look like below.



```
DataTable.ImportSheet "D:\UFT\Login.xls", 1, 1
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("username").Set "adactin123"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebEdit("password").SetSecure "511af1ac419e5b62da256061e4f79a56732df2216511ba827c77"
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").WebButton("Login").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").Link("Logout").Click
Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio").Link("Click here to login again").Click
```

Figure 16-3 – ImportSheet step in Editor View

Scenario 2: Suppose an application has 3 navigation buttons on each and every page. Let the buttons be "Cancel", "Back," and "Next". Now recording action on these buttons would add 3 objects per page in the repository. For a 10 page flow, this would mean 30 objects, which could have been represented just by using 3 objects. So instead of adding these 30 objects to the repository, we can just write 3 descriptions for the object and use it any page.

4. Modification to a test case is needed, but the Object Repository is read only or in shared mode i.e. changes may affect other scripts as well.
5. When you want to take action on similar types of objects i.e. suppose we have 20 textboxes on the page, and their names are in the form txt_1, txt_2, txt_3 and so on. Now adding all 20 the object repository would not be a good programming approach.

19.3 Descriptive Programming Syntax

One of the possible solutions to deal with dynamic objects is to use descriptive programming.

There are numbers of ways of using descriptive programming to define test objects like using name/value pairs, using description object or using child objects. Most common amongst them is using name/value pairs method.

Note: Using description object or child objects is out of scope of this book. Bu,t more information on this can be found on UFT help guide.

You can describe an object directly in a statement by specifying property: =value pairs, describing the object instead of specifying an object's name.

The general syntax is:

TestObject("PropertyName1:=PropertyValue1", "..." , "PropertyNameX:=PropertyValueX")

TestObject—the test object class could be WebEdit, WebRadioGroup etc….

PropertyName:=PropertyValue—the test object property and its value.

Each property: =value pair should be separated by commas and quotation marks. Note that you can enter a variable name as the property value, if you want to find an object based on property values you retrieved during a run session.

19.4 Handling Dynamic Object Using Descriptive Programming

Based on the example shown to us in previous section, let us try to resolve the problem using descriptive programming.

22

Integration with Quality Center

HP Quality Center/Application Lifecycle Management (QC/ALM) is a test management tool, which acts as a central repository for testing artifacts and helps to manage the testing process right through from releases to defects tracking. QC/ALM provides an intuitive and efficient method, for scheduling and running tests, collecting results, analyzing the results, and managing test versions. It also features a system for tracking defects, enabling you to monitor defects closely from initial detection until resolution.

UFT integrates pretty well with QC/ALM. We can use QC/ALM as a repository for storing UFT scripts and also for batch execution of scripts and storing test results.

Example

We have used the combination of QC/ALM with UFT at number of our client places. Most recently we had setup an automation framework for a client having a learning management system. We created an automation framework for them, stored all our UFT scripts in QC/ALM and executed the scripts from QC/ALM. This made the knowledge transfer to the client team very simple, as they knew that all their scripts and test results are within QC/ALM. All they needed to do was to click a button to execute the UFT scripts from QC/ALM.

As part of this chapter we will cover how we can store UFT scripts in QC/ALM and how to execute automation scripts from QC/ALM.

Key Objectives

- UFT – QC/ALM integration setup
- Saving UFT script in QC/ALM from UFT
- Launching UFT script from QC/ALM
- UFT script execution from QC/ALM

22.1 UFT – QC/ALM Integration Setup

In order to execute UFT scripts from QC/ALM on a particular machine, we need the following components setup on the machine where we want to use both UFT and QC/ALM

1. We should be able to access QC/ALM client on that machine (assuming that you already have QC/ALM server installed).
2. We need to have UFT installed on that machine.
3. We need to install UFT add-ins for QC/ALM on that machine. Although there is backward compatibility of UFT with older versions of QC/ALM, it is advisable to install and integrate UFT and QC/ALM of same versions. You will find UFT add-in for QC/ALM in the UFT installation itself.
   a. You will find the add-ins once you start UFT Setup file.

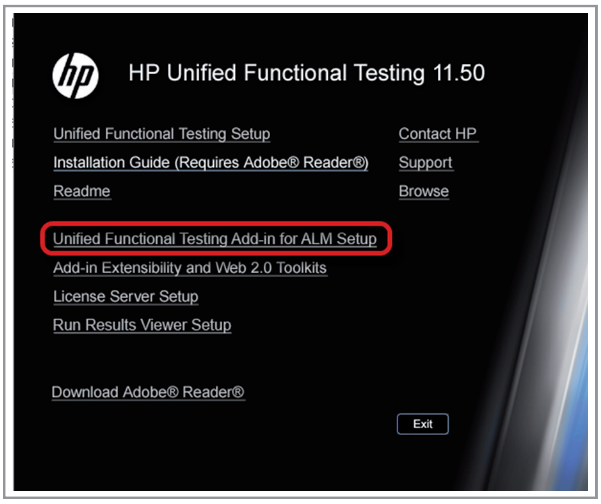
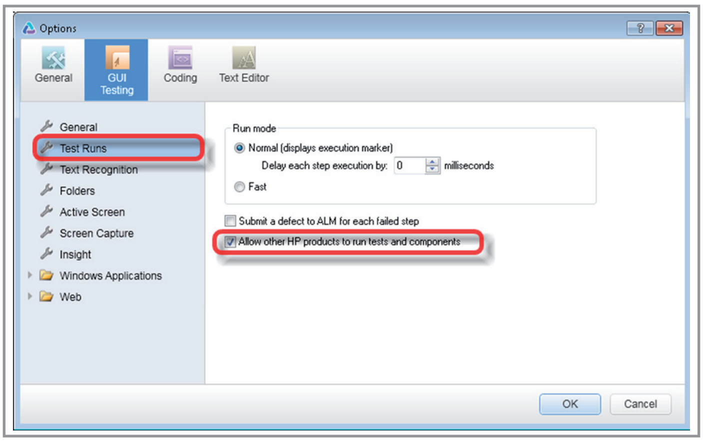

HP Unified Functional Testing 11.50
Unified Functional Testing Setup
Installation Guide (Requires Adobe® Reader®)
Readme
Contact HP
Support
Browse
Unified Functional Testing Add-in for ALM Setup
Add-in Extensibility and Web 2.0 Toolkits
License Server Setup
Run Results Viewer Setup
Download Adobe® Reader®
Exit

Figure 22-1 – UFT Add-in for ALM Setup

Note: Make sure you have admin rights to install this add-in on your machine.

4. Once you install this add-ins, Launch UFT.
5. Go to **Tool** –> **Options** → **GUI Testing** → **Test Runs**. Make sure that checkbox "**Allow other HP products to run tests and components**" is checked. Close the dialog after selection.



Run mode
(●) Normal (displays execution marker)
Delay each step execution by: 0 milliseconds
( ) Fast
☐ Submit a defect to ALM for each failed step
☑ Allow other HP products to run tests and components

Figure 22-2 – Allow other HP products to run tests and components

25

What's new with UFT 11.5

This chapter lists the new features in UFT 11.5, which did not exist in previous versions of QuickTest. This section is divided into General UFT enhancements which are generic to UFT and new features based on GUI tests. New API testing based features are out of scope of this book and not included in this chapter.

25.1 General UFT

One Composite Unified Tool for Application Testing

Unified Functional Testing (UFT) helps to test both the applications (GUI) layer and the business (API/Web Services) layer of your application all in one product.

UFT includes all of its "predecessors' (QuickTest and Service Test) capabilities along with new features.

If you use HP ALM (Application Lifecycle Management), then you can also create and run business process tests, from within UFT in addition to creating and editing individual GUI and API business components.

New IDE

The new IDE now has enhanced editor and coding capabilities

- Improved statement completion
- Customized and built-in code snippets
- Class and function browsers
- Go To dialog box

MDI: Edit multiple Testing Documents Simultaneously

In UFT, you can open and work with multiple documents stored in the same solution, including GUI or API tests, individual actions, business components, function libraries, and code files.

As you navigate from one document to another in the UFT document pane, relevant panes update to display the relevant data, and the Solution Explorer synchronizes to show you the currently active.

This is a great enhancement from previous versions of QTP, where users could not open multiple tests at the same time.

Error pane

The Errors pane displays a list of errors generated when opening, working with, saving, or running tests, components, function libraries, and user code files.

The Errors pane reports the following types of errors:

- Code syntax errors
- Missing resources (GUI testing only)
- Missing references (API testing only)
- Missing property values (API testing only)

Code syntax errors in previous versions of QTP were shown in Information pane, which is now removed. Also missing resources pane has been removed and replaced with composite error pane.

Automatically export test results

You can instruct UFT to automatically export run results in HTML or PDF format after every run session. Select **Tools > Options > General** tab **> Run Sessions** to set your preferences for the export.

In previous versions of QTP, the user had to manually export the results to HTML or pdf format.

Select License Types From Within the Add-in Manager Dialog box

If your concurrent license server has more than one valid license type installed, you can select the license you want to use in the add-in Manager dialog box. You have the option to choose a different license each time you open UFT (By default, your selection from the previous session is used).

25.2 GUI Testing

Insight: Image-based Object Identification

Along with object properties based recognition UFT also introduces new image-based identification (Insight), using which UFT recognizes user interface controls instead of object properties. This enables UFT to perform basic steps such as clicking, dragging, and dropping controls in applications that could not previously be tested. You can even use insight to test applications that run remotely on non-windows operating systems.

28

Common UFT Issues and FAQs

28.1 UFT Issues and FAQs

In order to keep this book simple and concise, there are quite a few topics we could not cover in detail. Keeping the larger audience of this book in mind, we have addressed a few of these topics and questions in this section.

What are the different ways to synchronize your scripts?

Many a times, your application performance will vary and this will need your UFT scripts speed to be appropriately manipulated.

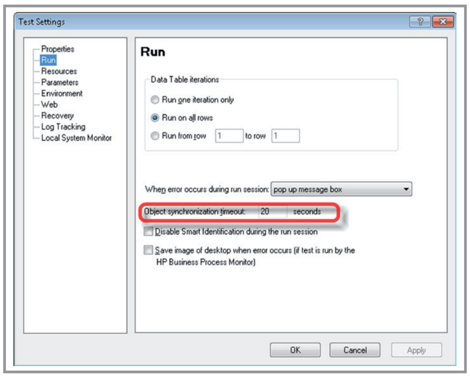
In one of the applications that we tested, it took more than 60 seconds for an application form to save and confirm the save. How does UFT assist in these situations?

There are 4 ways to handle synchronization in UFT

- Use Global Synchronization Timeout
- Use Static Wait Statements
- Use Dynamic Synchronization Point
- Use .Exist property

<u>Global Synchronization Timeout</u>

If you go to File → Settings → Run you will notice there is an Object Synchronization Timeout defined with default value of 20 seconds.



Run

Data Table iterations

Run one iteration only [✓ Run on all rows] Run from row 1 to row 1

When error occurs during run session: pop up message box

Object synchronization timeout: 20 seconds

☐ Disable Smart Identification during the run session

☐ Save image of desktop when error occurs (if test is run by the HP Business Process Monitor)

Figure 28-1 – Global Synchronization Timeout

How to use With statement in UFT script?

The **With** statement allows you to perform a series of statements on a specified object without requalifying the name of the object, which means do not need to repeat the object statements for each and every step.

For example

```
With Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio")
.WebEdit("username").Set "adactin123"
.WebEdit("password").SetSecure "4ffe76b5bf81a5c0168c57c97e46"
.WebButton("Login").Click
End With
```

In the above example we do not need to repeat object statement *Browser("AdactIn.com - Hotel Reservatio").Page("AdactIn.com - Hotel Reservatio")* for every step.

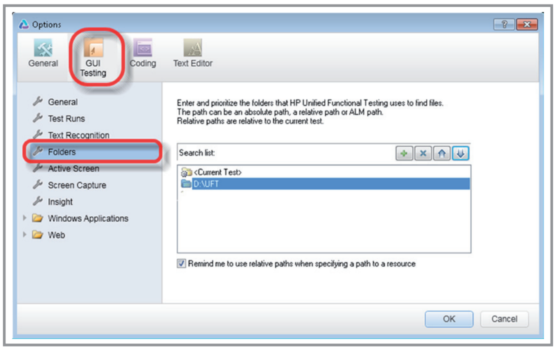
Note: You can nest With statements by placing one With block within another. However, because members of outer With blocks are masked within the inner With blocks, you must provide a fully qualified object reference in an inner With block to any member of an object in an outer With block.

Why and how to use relative path in Unified Functional Testing

As a good development practice, we should never hard-code paths in our script. For instance if you need to access your function library file, we should not use absolute paths like "D:\UFT\FunctionLibary\HotelApp.qfl". The key reason for that being that if you need to run the same script on another machine which does not have D: drive, it would be a huge maintenance effort to change the path in scripts every now and then.

The best solution is to keep the relative path in the script like "HotelApp.qfl", instead of specifying the complete absolute path.

You can define search path by going to **Tools → Options → GUI Testing → Folders**



Enter and prioritize the folders that HP Unified Functional Testing uses to find files.
The path can be an absolute path, a relative path or ALM path.
Relative paths are relative to the current test.

Search list:
<Current Test>
D:\UFT

[✓] Remind me to use relative paths when specifying a path to a resource

Figure 28-6 – Folders options

This suppresses only the results window, but a log will be created and can be accessed if required from **View → Last Run Results** menu

How to open an application using VBScript in UFT Editor View?

Use SystemUtil object to open and close applications and processes during a run session.

A SystemUtil.Run statement is automatically added to your test when you run an application from the Windows Start menu or the Run dialog box while recording a test. If it is not added, you can add it manually and specify the path of the .exe file to be executed as an argument. Below is a very common example to open Internet Explorer

Example:

SystemUtil.Run "C:\Program Files\Internet Explorer\iexplore.exe","www.google.com","C:\Documents and Settings\Administrator","open"

How do we handle run-time errors or exceptions in UFT?

There are two ways how run-time errors can be handled. We can create a recovery scenario and link it to every script or we can use ***On Error Resume next*** statement.

On Error Resume Next causes execution to continue to the statement immediately following the statement that caused the run-time error, or with the statement immediately following the most recent call out of the procedure containing the "On Error Resume Next" statement. This allows execution to continue despite of a run-time error.

28.2 Common Automation Interview Questions

I have been on interview panels for quite a few IT consulting companies and IT departments of Non-IT companies. In my personal experience, questions are centred around experience based, scenario based and approach based techniques, which helps the interviewer judge the depth of automation experience that a candidate possesses.

These are some of the questions I recommend that one needs to be prepared for:

- Can you explain the automation framework you have developed using QTP/UFT in your recent project?

Answer - Refer to our automation frameworks chapters to answers this question

- What are the key challenges you think you faced doing automation?

Answer - Some of the key challenges are

  * Automation Environment unavailability.
  * Application is unstable.
  * Features and workflows being changed frequently leading to maintenance issues.

TEST AUTOMATION USING UNIFIED FUNCTIONAL TESTING

- Test Automation using HP Unified Functional Testing (UFT) is the first book globally on HP UFT, latest version of HP test automation tool Quick Test Professional. This new UFT version is combination of HP QTP and HP Service Test, two separate products earlier from HP.
- Key audience for this book are functional testers and beginner test engineers who want to **learn UFT quickly** and want to learn it the right way which will help them create effective automation framework.
- This book has been designed with the objective of **simplicity and ease of understanding**.
- This book follows a **unique training based approach** instead of regular text book approach.
- As **a step by step guide**, it guides the student with every step on how to complete exercises with help of snapshots.
- Another major highlight of this book is that we are using our **custom developed web based application** instead of default HP sample Flight Reservation application.

Quotes from Reviewers

"A great hands-on guide on mastering Automation using the new Unified Functional Testing (UFT) Automation tool. An example custom built web based application used throughout the book ensures a quick grasp of key concepts. This book, with it's real life examples, is sure to benefit the novice beginner as well as seasoned Automation professionals looking to step up to UFT 11.5, highly recommended!"

- S. Constâncio Colaço (Senior QA Analyst/Consultant)

"Outstanding book on QuickTest Professional. Rather than a text book approach, a training based approach to explain UFT features is an amazing concept. Also learning the tool over web based application is fabulous as most organizations have complete web based application environment"

- Philip Smith (Senior Automation Analyst)

Navneesh Garg is a recognized automation architect and corporate trainer, certified in HP QTP, HP Quality Center, HP LoadRunner, IBM Rational Functional Tester and a Certified Ethical Hacker.

He is a co- founder of several successful IT companies like Adactin Group, CresTech Software and Planios Technologies.

As a corporate trainer and test consultant, he has worked with clients like Macquarie Bank, Corporate Express, Max New York Life Insurance, Accenture, NSW Road and Maritime Services, Australian Dept of Education, HCL Technologies, Sapient, Fidelity group, Adobe Systems, and many more. He has training experience in diverse geographies including Australia, India, Hong Kong, and USA

As a tool specialist he has worked on variety of functional automation tools including Selenium, TestComplete, TestPartner, SilkTest, Watir, RFT, HP QTP/UFT on varied technologies included Web, Java, Dot-net, SAP, Peoplesoft and Seibel.

He can be reached at:
Navneesh.garg@adactin.com