

Code [Dada](#)

QTP [Material](#)

**Prerequisites:**

**a) Software Testing Principles & Practices**

**b) Programming Fundamentals**

**c) Database Fundamentals**

\*\*\*\*\*

**Introduction & QTP Test Process**

---

- **Overview on Test Automation & Types of Test Tool**

- § Drawbacks of Manual Testing
- § Advantages of Test Automation
- § Drawbacks of Test Automation
- § Types of Test Tool (Business Classification)
- § Types of Test Tool (Technical Classification)

- **Basic Features of QTP**

- § Supporting Environments
- § Version History
- § Types of License
- § Built-in features
- § Integrated Tools (from HP)
- § Integrated Tools (Others)

- **QTP Test Process**

- § Planning
- § Generating Basic Tests
- § Enhancing Tests

- § Debugging Tests
- § Running Tests
- § Analyzing Test Results
- § Reporting Defects

### Key Elements of QTP, Recording & Running

---

#### o **Key Elements of QTP**

- § Test Pane
- § Active Screen
- § Data Table
- § Debug Viewer
- § Missing Resources
- § Information
- § QTP Commands

#### o **Recording & Running Tests**

- § Recording Process
- § Running Process
- § Recording Modes
- § Advantages of Recording
- § Drawbacks of Recording

### Object Repository

---

#### **Software objects**

##### **Types of object in QTP**

- § Runtime Objects
- § Test Objects
- § Utility Objects
- § Automation Objects

## **Object Repository**

- § Local Repository
- § Shared Repository

## **Operations on Object Repository**

- § Adding Objects
- § Renaming Objects
- § Deleting Objects
- § Associating Object Repositories to an Action
- § Exporting Local Objects
- § Merging Repositories
- § View Options
- § Spying Objects
- § Importing and Exporting Repositories using XML
- § Defining New Test Objects

- **Keyword driven methodology**

- § Generating Tests Manually through Expert view
- § Generating Steps through Keyword view
- § Generating statements through Step Generator

## **Object Identification Configuration**

### **Object Identification Configuration**

- Normal Identification
- Mandatory Properties
- Assertive Properties
- Smart Identification
- Base filter Properties
- Option filter properties
- Ordinal Identifier
- Making Tool Settings Global

### **Virtual Object Configuration**

- § Virtual Object Manager
- § Defining a Virtual Object

- § Removing or Disabling Virtual Object Definitions

## Descriptive Programming

---

### Descriptive Programming

- § Advantages of Descriptive Programming
- § Static Programming
- § Dynamic Programming
- § Creating Library files using Constants /Collection objects
- § Handling Dynamic Objects
- § Handling duplicate Objects
- § Retrieving Child Objects

### Methods/Operations

(Activate Click, Close, dblclick, Set, SetSecure, Select, GetVisibleText, GetRoProperty, GetContent, GetItemsCount, GetItem, Sync, Navigate, WaitProperty and Others...)

## Checkpoints & Output Values

### Inserting Check points

- § Standard Checkpoint
- § Text Checkpoint
- § Text Area Checkpoint
- § Bitmap Checkpoint
- § Database Checkpoint
- § Accessibility Checkpoint
- § XML Checkpoint (From Application)
- § XML Checkpoint (From Resource)
- § Page Checkpoint
- § Image Checkpoint
- § Table Checkpoint

### Inserting Output values

- § Standard Output value
- § Text Output value
- § Text Area Output value
- § Database Output value
- § XML Output value (From Application)
- § XML Output value (From Resource)

## Parameterization, Synchronization

### Synchronization

- § Inserting Wait Statement
- § Inserting Synchronization Point
- § Increasing Tool Default Synchronization Time
- § Sync method (only for web)
- § Selecting appropriate method

### Parameterization

- § Data Driven Testing through Loop statements
- § Dynamic Test Data Submission through Looping
- § Data Driven Testing through Data Table
- § Importing Test Data from External files (Text and Excel Files)
- § Creating a Database, Tables and Entering Data
- § Creating DSN
- § Importing Test Data from Databases
- § Data Driven Testing using Data Table Methods
- § Data Driven Testing by fetching the Test Data directly from a Text file
- § Data Driven Testing by fetching the Test Data directly from an Excel file
- § Data Driven Testing by fetching the Test Data directly from a Database

## Actions, Environment Variables

### Actions

- Types of actions

§ Non-reusable Actions

§ Re-Usable actions

§ External Actions

- Creating an Action
- Splitting Actions
- Renaming an Action
- Deleting an Action
- Making an Reusable/Non-Reusable
- Calling an existing Action
- Copying an Actions
- Action Parameters

### **Environment Variables**

Built-in Variables

User Defined Variables

Defining, modifying and associating environment variables

Loading Environment file during run-time

### **Transaction Points, Adding Comments**

#### **Inserting Transaction Points**

§ Start Transaction & End Transaction

§ Timer Function

§ Defining Test Results

#### **Adding Comments**

§ Purpose of Comments

§ Basic Syntax

§ Comment a Block of Statements

§ Uncomment a Block

### **Debugging Tests, Batch Testing**

#### **Debug Commands & Break Points**

§ Step Into

§ Step Over

§ Step Out

- o Inserting/Removing Brake Points
- o Watching variables
- o Changing Variable values

### **Batch Testing**

- Test Batch Runner
- Creating Test Batches
- Providing permission to Test Batch Runner
- Running Test Batches
- Running Partial Batches

### **Automation Object Model**

Automating QuickTest Operations

### **Recovery Scenarios, Quality Center**

#### **Recovery Scenarios**

##### **Trigger events**

Pop-Up window

Object State

Test Run Error

Application Crash

##### **Recovery Operations**

Keyboard or Mouse Operation

Close Application Process

Function Call

Restart Microsoft Windows

Creating Recovery Scenarios

Associating Recovery Scenarios

#### **Quality Center (Site Administration)**

- § Creating Domains/ Deleting domains
- § Creating a project/ deleting a project
- § Creating users
- § Deleting users

§ Assigning users to a project

§ Tracking a project status

## **Quality Center**

### **Quality Center**

#### **a) Requirements**

Creating/Modifying/Deleting test requirements and child requirements

#### **b) Test plan**

Creating/Modifying/Viewing/Deleting manual as well as automated tests

Mapping requirements with test cases

Exporting Manual test cases to quality center project

Exporting QTP tests to quality center project

#### **c) Test Lab**

Creating folders

Forming test batches

Executing manual and automated tests

#### **d) Defects**

Posting defects

Tracking defects

Sending defects directly from QTP result window

## **VB Script**

### **Introduction, Variables**

#### **VB Script Fundamentals & Features**

- Scripting Languages vs. Programming Languages
- Basic Features of VBScript
- Hosting Environments & Script Engines
- Crating & Running a Script

#### **Data Types, Variables**

- VBScript data types
- Declaring Variables and Option Explicit Statement
- Assigning Values to Variables



- Scalar Variables and Array Variables
- Dynamic Arrays, Dimensional Arrays

### Comments, Constants, Operators

#### Comments

#### Constants

#### Operators

- Operator Precedence
- Arithmetic Operators
- Comparison Operators
- Concatenation Operators
- Logical Operators

### Flow Control (Conditional Statements)

- VBScript Conditional Statements
- Running a statement if condition is true (Single line Syntax)
- Running a block of Statements if Condition is true
- Running Certain Statements if a Condition is True and Running Others if a Condition is False
- Deciding Among Several Alternatives (If..Else If...)
- Executing a certain block of statements when two / more conditions are True (Nested If...)
- Deciding among several Alternatives (Select Case Statement)

### Flow Control (Looping through the Code)

- Do...Loop (Loops while or until a condition is True.)
- While...Wend (Loops while a condition is True.)
- For...Next: Uses a counter to run statements a specified number of times.
- For Each...Next (Repeats a group of statements for each item in a collection or each element of an array.)
- Nested Loops

### User Defined Functions

- Types of Procedures
- Internal & External Functions
- Creating Library files
- Creating Sub Procedures
- Creating Function Procedures
- Associating Library files
- Loading Function Libraries run-time
- Generic Functions
- Project Specific Functions

### Built-In Functions

- Conversions (Cdbl, Cint etcc...)
- Dates/Times (Date, Time, Now, IsDate, DateDiff, etc...)
- Formatting Strings

- Input/Output (InputBox and MsgBox)
- Math
- Miscellaneous
- Rounding
- Strings (Len, Left, Right, Mid Etc....)
- Arrays (Array, IsArray, Split etc..)
- Variants

## File System Operations

### File System Operations

- Computer File System
- OS Distribution
- FileSystemObject

#### Examples:

- Create a Folder
- Check if the Folder Exist or not? If not create the Folder
- Copy a Folder
- Delete a folder
- Return a Collection of Disk Drives
- Get available space on a Drive
- Creating a Text File
- Check if the File Exist or not? If not create the File
- Read Data Character by Character from a text file
- Read Line by Line from a Text File
- Data Driven Testing by fetching Test data directly from a Text file.
- Write Data to a Text File
- Delete a Text File
- Check if the File Exists or not? If Exists delete the File
- Calculate size of a Text File
- Compare Two Text File by Size, by Text and by Binary values
- Count the number of times a word appears in a Text File
- Capture all Button Names and Export to a Text File
- Capture Customer Names from 1 to 10 Orders in FR and export to a Text File

## Excel Object Model

### Excel Object Model:

Creating an Automation Object in Excel.Application class

#### Excel Scripting Examples:

- Create an Excel file, enter some data and save the file through VB script?
- Check if the Excel file exists or not, if exists open the file and enter some data, If not Exists create the file and enter some data and save the file through VB script?
- Data Driven Testing through an External Excel Sheet
- Data Driven Testing using Data Table methods
- Calculator Example using Data Table methods

- Open an Excel Spreadsheet
- Read an Excel Spreadsheet
- Add Formatted Data to a Spreadsheet
- Sort an Excel Spreadsheet on Three Different Columns
- Short an excel sheet column
- Add New Sheet to Excel File
- Rename Sheets in an Excel File (WorkBook)
- Add a Sheet to an Excel File (WorkBook) and change the Position

### Database Object Model

- o ActiveX Data Objects
- o ADODB Connection Object & Recordset Object
- o Test Database Creation & Collecting Test Data
- o Databases Connections (Ms-Access, SQL Server and Oracle)
- o Data Driven Testing by fetching Data from a Database
- o Data Comparisons

### Regular Expressions

- Understanding & Using Regular Expressions
- Handling Dynamic Objects
- Using Regular Expressions in Checkpoints
- Defining Regular Expressions
- Search Operations
- Regular expressions in Recovery scenarios
- Regular Expression Object

### Error Handling

- Overview on Error Handling
- Error Handling Methods in QTP
- Synchronization
- Exist Property
- Exit Statement
- On Error Statement
- Recovery scenarios

### GUI Scripting

- Handling GUI Objects
- Object State Verification
- Input Domain Coverage
- Output Domain Coverage
- GUI Script Examples

### Web Scripting

- Checking Links
- Testing Forms
- Output Domain Coverage
- Cookies Testing
- Web Script Examples

### Introduction, Basic Framework

- Automation Framework, Types of Framework
- Creating Folder Structure
- Creating & Storing Automation Resources
- Organizing & Managing Resources
- Generating Driver Scripts
- Generating Initialization Script (AOM Script)

### Data Driven Framework

- Test Data Collection
- Ways of Parameterization
- Data Driven Implementation Techniques
- Using Data Table Methods

### Keyword Driven Framework

- Keywords
- Keyword Driven Approach
- Creating Functions
- Creating Test Automation resources
- Creating Organizer Spread sheet
- Generating the Driver Script
- Generating the Initialization Script

### Hybrid Framework

- Hybrid Automation Framework
- Keyword Driven & Data Driven Test Approach
- Advantages of Hybrid Approach
- Organizing & Managing Resources

Posted in [Uncategorized](#) | Tagged [qtp](#), [qtp online training](#), [qtp scripts](#), [qtp training](#) | [Leave a Comment »](#)

### QTP Script Examples

November 14, 2009 by [Naseer](#)

**1) Verify Login Boundary** (Check all the boundary conditions of the Login dialog box. Checks to see if the correct message appears in the error window (Flight Reservation Message)

```
ApplicationDir = Environment("ProductDir")
```

```
ApplicationPath = "\samples\flight\app\flight4a.exe"
```

```

If Window("Flight Reservation").Exist(2) Then

Window("Flight Reservation").Close

SystemUtil.Run ApplicationDir & ApplicationPath

Elseif Not Dialog("Login").Exist(1) Then

SystemUtil.Run ApplicationDir & ApplicationPath

End If

Dialog("Login").WinEdit("Agent Name:").Set Datatable.Value ("AgentName",dtGlobalSheet)

Dialog("Login").WinEdit("Password:").Set Datatable.Value ("Password",dtGlobalSheet)

Dialog("Login").WinButton("OK").Click

If Dialog("Login").Dialog("Flight Reservations").Exist(1) and Datatable.Value ("Status",dtGlobalSheet)="Fail" Then

Dialog("Login").Dialog("Flight Reservations").Static("Agent name must be at").Check CheckPoint("Agent name must be at least 4 characters long.")

Dialog("Login").Dialog("Flight Reservations").WinButton("OK").Click

Elseif Window("Flight Reservation").Exist(10) and Datatable.Value ("Status",dtGlobalSheet)="Pass" Then

Reporter.ReportEvent PASS,"Login: ", "Succeeded"

Else

Reporter.ReportEvent Fail,"Login: ", "Combination #" & Datatable.GetCurrentRow & " was not according to Excel file"

End If

```

**2) Verify Cancel Operation** (in Login Dialog box, if user selects cancel button, before enter any data after enter data dialog box should be disappeared.)

- 1) Invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 2) Dialog("Login").Activate
- 3) Dialog("Login").WinButton("Cancel").Click
- 4) If Dialog("Login").Exist (2) =True Then
- 5) Reporter.ReportEvent 1,"sd", "Fail"
- 6) Else
- 7) Reporter.ReportEvent 0,"sd", "Pass"
- 8) Invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

- 9) End If
- 10) Dialog("Login").Activate
- 11) Dialog("Login").WinEdit("Agent Name:").Set "asdf"
- 12) Dialog("Login").WinButton("Cancel").Click
- 13) If Dialog("Login").Exist (2) =True Then
- 14) Reporter.ReportEvent 1,"sd","Fail"
- 15) Else
- 16) Reporter.ReportEvent 0,"sd","Pass"
- 17) Invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 18) End If
- 19) Dialog("Login").Activate
- 20) Dialog("Login").WinEdit("Agent Name:").Set "asdf"
- 21) Dialog("Login").WinEdit("Password:").SetSecure "4a993af45dcdbd506c8451b274d2da07b38ff5531"
- 22) Dialog("Login").WinButton("Cancel").Click
- 23) If Dialog("Login").Exist (2)=True Then
- 24) Reporter.ReportEvent 1,"sd","Fail"
- 25) Else
- 26) Reporter.ReportEvent 0,"sd","Pass"
- 27) Invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 28) End If
- 29) Dialog("Login").Activate
- 30) Dialog("Login").WinEdit("Agent Name:").Set "asdf"
- 31) Dialog("Login").WinEdit("Password:").SetSecure "4a993af45dcdbd506c8451b274d2da07b38ff5531"
- 32) Dialog("Login").WinButton("OK").Click

### **3) Verify Addition, Subtraction, Multiplication and Division Operations in Calculator Application.**

- 1) Dim aRes,sRes,dRes,mRes
- 2) VbWindow("VbWindow").Activate
- 3) VbWindow("VbWindow").VbEdit("VbEdit").Set "10"

```
4)  VbWindow("VbWindow").VbEdit("VbEdit_2").Set "20"

5)  v1=VbWindow("VbWindow").VbEdit("VbEdit").GetROProperty ("text")

6)  v2=VbWindow("VbWindow").VbEdit("VbEdit_2").GetROProperty ("text")

7)  VbWindow("VbWindow").VbButton("ADD").Click

8)  aRes=VbWindow("VbWindow").VbEdit("VbEdit_3").GetVisibleText

9)  VbWindow("VbWindow").VbButton("SUB").Click

10) sRes=VbWindow("VbWindow").VbEdit("VbEdit_3").GetVisibleText

11) VbWindow("VbWindow").VbButton("MUL").Click

12) mRes=VbWindow("VbWindow").VbEdit("VbEdit_3").GetVisibleText

13) VbWindow("VbWindow").VbButton("DIV").Click

14) dRes=VbWindow("VbWindow").VbEdit("VbEdit_3").GetVisibleText

15) v1=cdbl(v1)

16) v2=cdbl(v2)

17) aRes=cdbl (aRes)

18) sRes=cdbl (sRes)

19) mRes=cdbl (mRes)

20) dRes=cdbl (dRes)

21) If aRes=v1+v2 Then

22) Reporter.ReportEvent 0,"Res","Addition Passed"

23) else

24) Reporter.ReportEvent 1,"Res","Addition Failed"

25) End If

26) If sRes=v1-v2 Then

27) Reporter.ReportEvent 0,"Res","Subtraction Passed"

28) else

29) Reporter.ReportEvent 1,"Res","Subtraction Failed"

30) End If

31) If mRes=v1*v2 Then
```

32) Reporter.ReportEvent 0,"Res","Multiplication Passed"

33) else

34) Reporter.ReportEvent 1,"Res","Multiplication Failed"

35) End If

36) If dRes=v1/v2 Then

37) Reporter.ReportEvent 0,"Res","Division Passed"

38) else

39) Reporter.ReportEvent 1,"Res","Division Failed"

40) End If

**4) Verify state of Update Order Button, before open an Order and after open an Order** (in Flight Reservation before opening an order Update Order button should be disabled after opening an order enabled.)

1) Option explicit

2) Dim bo,ao

3) If Not window("Flight Reservation").Exist (2) Then

4) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

5) Dialog("Login").Activate

6) Dialog("Login").WinEdit("Agent Name:").Set "Naseer"

7) Dialog("Login").WinEdit("Password:").SetSecure "4aa8bce9984f1a15ea187a2da5b18c545abb01cf"

8) Dialog("Login").WinButton("OK").Click

9) End If

10) Window("Flight Reservation").Activate

11) bo=Window("Flight Reservation").WinButton("Update Order").GetROProperty ("Enabled")

12) Window("Flight Reservation").WinButton("Button").Click

13) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No:").Set "ON"

14) Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "1"

15) Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

16) ao=Window("Flight Reservation").WinButton("Update Order").GetROProperty ("Enabled")

17) If bo=False Then



18) Reporter.ReportEvent 0,"Res","Update Order Button Disabled"

19) else

20) Reporter.ReportEvent 1,"Res","Update Order Button Enabled"

21) End If

22) If ao=True Then

23) Reporter.ReportEvent 0,"Res","Update Order Button Enabled"

24) else

25) Reporter.ReportEvent 1,"Res","Update Order Button Disabled"

26) End If

**5) Price Consistency, In Flight Reservation** (In Flight Reservation, First class price=3\*Economy class price and Business class price=2\*Economy class price)

1) Option explicit

2) Dim n,f,b,e

3) If Not window("Flight Reservation").Exist (2) Then

4) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

5) Dialog("Login").Activate

6) Dialog("Login").WinEdit("Agent Name:").Set "asdf"

7) Dialog("Login").WinEdit("Password:").SetSecure "4aa8b7b7c5823680cfcb24d30714c9bbf0dff1eb"

8) Dialog("Login").WinButton("OK").Click

9) End If

10) For n= 1 to 10 step 1

11) Window("Flight Reservation").Activate

12) Window("Flight Reservation").WinButton("Button").Click

13) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No:").Set "ON"

14) Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set n

15) Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

16) Window("Flight Reservation").WinRadioButton("First").Set

17) f=Window("Flight Reservation").WinEdit("Price:").GetVisibleText

```

18) Window("Flight Reservation").WinRadioButton("Business").Set
19) b=Window("Flight Reservation").WinEdit("Price:").GetVisibleText
20) Window("Flight Reservation").WinRadioButton("Economy").Set
21) e=Window("Flight Reservation").WinEdit("Price:").GetVisibleText
22) f=cdbl(mid(f,2,len (f-1)))
23) b=cdbl(mid(b,2,len (b-1)))
24) e=cdbl(mid(e,2,len (e-1)))
25) If f=3*e and b=2*e Then
26) Reporter.ReportEvent 0,"Res","Pricy Consistancy is there"
27) else
28) Reporter.ReportEvent 1,"Res","Pricy Consistancy is NOT there"
29) End If
30) Window("Flight Reservation").WinButton("Button_2").Click
31) Window("Flight Reservation").Dialog("Flight Reservations").WinButton("No").Click
32) Next

```

**6) Verify Total, In Flight Reservation** (In Flight Reservation, Total = Tickets \* Price)

```

1) Option Explicit
2) Dim t,p,tot,n
3) For n= 1 to 10 step 1
4) If Not window("Flight Reservation").Exist (2) Then
5) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe","","C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\","open"
6) Dialog("Login").Activate
7) Dialog("Login").WinEdit("Agent Name:").Set "Naseer"
8) Dialog("Login").WinEdit("Password:").SetSecure "4aa892d62c529f1c23298175ad78c58f43da8e34"
9) Dialog("Login").WinButton("OK").Click
10) End If
11) Window("Flight Reservation").Activate

```

```

12) Window("Flight Reservation").WinButton("Button").Click

13) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

14) Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set n

15) Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

16) t=Window("Flight Reservation").WinEdit("Tickets:").GetVisibleText

17) p=Window("Flight Reservation").WinEdit("Price:").GetVisibleText

18) tot=Window("Flight Reservation").WinEdit("Total:").GetVisibleText

19) t=cdbl (t)

20) p=Cdbl(mid(p,2,len (p-1)))

21) tot=Cdbl(mid(tot,2,len (tot-1)))

22) If tot=t*p Then

23) Reporter.ReportEvent 0,"Res","Calculation Passed"

24) else

25) Reporter.ReportEvent 1,"Res","Calculation Failed"

26) End If

27) Next

```

**7) Verify Flight From & Flight To Combo Boxes** (In Flight reservation, select an item from Fly From: combo box and verify weather that item available or not in Fly To: combo box, like this select all items one by one in Fly From and verify weather selected items available or not in Fly To.)

```

1) Option explicit

2) Dim qtp,flight_app,f,t,i,j,x,y

3) If Not Window("text:=Flight Reservation").Exist (7)= True Then

4) QTP=Environment("ProductDir")

5) Flight_app="\samples\flight\app\flight4a.exe"

6) SystemUtil.Run QTP & Flight_app

7) Dialog("text:=Login").Activate

8) Dialog("text:=Login").WinEdit("attached text:=Agent Name:").Set "asdf"

9) Dialog("text:=Login").WinEdit("attached text:=Password:").SetSecure "4aa5ed3daf680e7a759bee1c541939d3a54a5b65"

10) Dialog("text:=Login").WinButton("text:=OK").Click

```

- 11) End If
- 12) Window("text:=Flight Reservation").Activate
- 13) Window("text:=Flight Reservation").WinButton("window id:=6").Click
- 14) Window("text:=Flight Reservation").ActiveX("acx\_name:=MaskedTextBox", "window id:=0").Type "090910"
- 15) f=Window("text:=Flight Reservation").WinComboBox("attached text:=Fly From:").GetItemCount
- 16) For i= 0 to f-1 step 1
- 17) Window("text:=Flight Reservation").WinComboBox("attached text:=Fly From:").Select (i)
- 18) x=Window("text:=Flight Reservation").WinComboBox("attached text:=Fly From:").GetROProperty ("text")
- 19) t=Window("text:=Flight Reservation").WinComboBox("attached text:=Fly To:","x:=244","y:=147").GetItemCount
- 20) For j= 0 to t-1 step 1
- 21) Window("text:=Flight Reservation").WinComboBox("attached text:=Fly To:","x:=244","y:=147").Select (j)
- 22) y=Window("text:=Flight Reservation").WinComboBox("attached text:=Fly To:","x:=244","y:=147").GetROProperty ("text")
- 23) If x <> y Then
- 24) Reporter.ReportEvent 0,"Res","Test Passed"
- 25) Else
- 26) Reporter.ReportEvent 1,"Res","Test Failed"
- 27) End If
- 28) Next
- 29) Next

**Verify Order No Entry in Flight Reservation.** (In Open Order dialog box, Order No object accepts numeric values only.)

- 1) If Not window("Flight Reservation").Exist (2) Then
- 2) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 3) Dialog("Login").Activate
- 4) Dialog("Login").WinEdit("Agent Name:").Set "asdf"
- 5) Dialog("Login").WinEdit("Password:").SetSecure "4aa9ccae3bb00962b47ff7fb0ce3524c1d88cb43"
- 6) Dialog("Login").WinButton("OK").Click
- 7) End If
- 8) Window("Flight Reservation").Activate

- 9) Window("Flight Reservation").WinButton("Button").Click
- 10) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
- 11) Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "a"
- 12) ord=Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").GetVisibleText
- 13) If ord= "a" Then
- 14) Reporter.ReportEvent 1,"Res","Order No Object is taking invalid data"
- 15) else
- 16) Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "1"
- 17) Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click
- 18) End If

#### **9) Get Test Data from a Flat file and use in Data Driven Testing (through Scripting)**

- 1) Dim fso,myfile
- 2) Set fso=createobject("scripting.filesystemobject")
- 3) Set myfile= fso.opentextfile ("F:\gcr.txt",1)
- 4) myfile.skipline
- 5) While myfile.atendofline <> True
- 6) x=myfile.readline
- 7) s=split (x, ",")
- 8) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 9) Dialog("Login").Activate
- 10) Dialog("Login").WinEdit("Agent Name:").Set s(0)
- 11) Dialog("Login").WinEdit("Password:").SetSecure s(1)
- 12) Dialog("Login").WinButton("OK").Click
- 13) Window("Flight Reservation").Close
- 14) Wend

#### **10) Get Test Data From a Database and use in Data Driven Testing (through Scripting)**

- 1) Dim con,rs

- 2) Set con=createobject("Adodb.connection")
- 3) Set rs=createobject("Adodb.recordset")
- 4) con.provider=("microsoft.jet.oledb.4.0")
- 5) con.open "C:\Documents and Settings\Administrator\My Documents\Gcr.mdb"
- 6) rs.open "Select \* From Login",con
- 7) While rs.eof <>True
- 8) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 9) Dialog("Login").Activate
- 10) Dialog("Login").WinEdit("Agent Name:").Set rs.fields ("Agent")
- 11) Dialog("Login").WinEdit("Password:").Set rs.fields ("Password")
- 12) Dialog("Login").WinButton("OK").Click
- 13) Window("Flight Reservation").Close
- 14) rs.movenext
- 15) Wend

**11) Count, how many links available in Mercury Tours Home Page.**

- 1) Set oDesc = Description.Create()
- 2) oDesc("micclass").Value = "Link"
- 3) Set Lists = Browser("Welcome: Mercury").Page("Welcome: Mercury").ChildObjects (oDesc)
- 4) NumberOfLinks = Lists.Count()
- 5) Reporter.ReportEvent 2,"Res","Number of Links are: "&NumberOfLinks

**12) Count, how many Buttons and Edit boxes available in Flight Reservation main window.**

- 1) If Not window("Flight Reservation").Exist (2) Then
- 2) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"
- 3) Dialog("Login").Activate
- 4) Dialog("Login").WinEdit("Agent Name:").Set "Naseer"
- 5) Dialog("Login").WinEdit("Password:").Set "mercury"
- 6) Dialog("Login").WinButton("OK").Click
- 7) End If

```

8) Set oDesc = Description.Create()

9) oDesc("micclass").Value = "WinButton"

10) Set Buttons = Window("text:=Flight Reservation").ChildObjects (oDesc)

11) Num_Buttons = Buttons.Count()

12) Set oDesc1=Description.Create()

13) oDesc1("micclass").Value="WinEdit"

14) Set Editboxes=Window("text:=Flight Reservation").ChildObjects (oDesc1)

15) Num_Editboxes= editboxes.count ()

16) sum= Num_Buttons+Num_Editboxes

17) Reporter.ReportEvent 2, "Res", "Total Buttons: "& Num_Buttons &"Total Edit boxes: "& Num_Editboxes

```

### **13) Verify search options in Open Order Dialog box**

(After selecting open order, 3 search options should be enabled and not checked,

After selecting Order No option, other options should be disabled,

After selecting Customer Name, Flight date option enabled and Order No disabled

After selecting Flight date option, Customer Name enabled and Order No disabled)

```

1) If Not window("Flight Reservation").Exist (2) Then

2) SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

3) Dialog("Login").Activate

4) Dialog("Login").WinEdit("Agent Name:").Set "Naseer"

5) Dialog("Login").WinEdit("Password:").SetSecure "4aa9ed25bc0ebde66ed726ad87d7e991347d8b9c"

6) Dialog("Login").WinButton("OK").Click

7) End If

8) Window("Flight Reservation").Activate

9) Window("Flight Reservation").WinButton("Button").Click

10) Window("Flight Reservation").Dialog("Open Order").Activate

11) oe=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").GetROProperty ("Enabled")

12) ce=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").GetROProperty ("Enabled")

13) fe=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").GetROProperty("Enabled")

```

```
14) oc=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").GetROProperty ("Checked")

15) cc=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").GetROProperty ("Checked")

16) fc=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").GetROProperty("Checked")

17) If (oe=true and ce=true and fe=true) and (oc="OFF" and cc="OFF" and fc="OFF") Then

18) Reporter.ReportEvent 0,"Res","Pass"

19) else

20) Reporter.ReportEvent 1,"Res","Fail"

21) End If

22) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

23) ono=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").GetROProperty ("Checked")

24) If Then

25) fd=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").GetROProperty ("Enabled")

26) ono=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").GetROProperty ("Enabled")

27) fd=false

28) ono=false

29) Reporter.ReportEvent 0,"Res","Pass"

30) else

31) Reporter.ReportEvent 1,"Res","Fail"

32) End If

33) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "OFF"

34) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").Set "ON"

35) cn=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").GetROProperty ("Checked")

36) If cn="ON" Then

37) ono=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").GetROProperty ("Enabled")

38) fd=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").GetROProperty ("Enabled")

39) fd=True

40) ono=false

41) Reporter.ReportEvent 0,"Res","Pass"
```



```

42) else

43) Reporter.ReportEvent 1,"Res","Fail"

44) End If

45) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").Set "OFF"

46) Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").Set "ON"

47) fd=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Flight Date").GetROProperty ("Checked")

48) If fd="ON" Then

49) ono=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").GetROProperty ("Enabled")

50) cn=Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").GetROProperty ("Enabled")

51) cn=True

52) ono=false

53) Reporter.ReportEvent 0,"Res","Pass"

54) else

55) Reporter.ReportEvent 1,"Res","Fail"

56) End If

```

**14) In Login Dialog box, Verify Help message** (The message is 'The password is 'MERCURY')

```

1)  If Not Dialog("Login").Exist (2) Then

2)  SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

3)  End If

4)  Dialog("Login").Activate

5)  Dialog("Login").WinButton("Help").Click

6)  message=Dialog("Login").Dialog("Flight Reservations").Static("The password is 'MERCURY'").GetROProperty("text")

7)  If message="The password is 'MERCURY'" Then

8)  Reporter.ReportEvent 0,"Res","Correct message "&message

9)  else

10) Reporter.ReportEvent 1,"Res","Worng message "

11) End If

```

**15) Count all opened Browsers on desktop and close them all?**

- 1) Set oDesc = Description.Create()
- 2) oDesc("micclass").Value = "Browser"
- 3) Set Browsers = Desktop.ChildObjects (oDesc)
- 4) NumberOfBrowsers = Browsers.Count()
- 5) Reporter.ReportEvent 2,"Res","Number of Browsers are: "&NumberOfBrowsers
- 6) For Counter=0 to NumberOfBrowsers-1
- 7) Browsers(Counter).Close
- 8) Next

**16) Create an Excel file, enter some data and save the file through VB scripting?**

- 1) Dim objexcel
- 2) Set objExcel = createobject("Excel.application")
- 3) objexcel.Visible = True
- 4) objexcel.Workbooks.add
- 5) objexcel.Cells(1, 1).Value = "Testing"
- 6) objexcel.ActiveWorkbook.SaveAs("f:\exceltest.xls")
- 7) objexcel.Quit

Basically Software Testing is 2 types

- 1) Manual Testing
- 2) Test Automation

**Manual Testing:**

Testing software manually is called Manual Testing. we can test all accepts of software manually.

Below Testing Types can be tested manually

Test Types:

- a) Functional Testing
- b) Regression Testing
- c) GUI Testing
- d) Usability Testing
- e) Security Testing
- f) Compatibility Testing
- g) Recovery Testing
- h) Reliability testing
- Etc...

#### **Drawbacks of Manual Testing**

- (i) Time consuming.
- (ii) More resources required.
- (iii) Human Errors
- (iv) Repetition of the Task is not much
- (v) Tiredness
- (vi) Simultaneous auctions are not possible (Parallel)

#### **Test Automation:**

Testing Software using any Automation tools is called Test Automation

Advantages of Test Automation:

- a) Fast: Tools are faster in execution than human users
- b) Reliable: Tools are reliable in complex calculations and tasks
- c) Reusable: we can reuse Automated Tests at any number of times
- d) Repeatable: we can repeat same operations for required number of times
- e) Programmable: we can use flow control statements for applying logic
- f) Comprehensive: we can execute test batches without human interaction also

#### **Test Automation can be used in below areas of Testing:**

- a) Functional & Regression Testing
- b) Load/Stress/Performance Testing
- c) Security Testing
- d) Unit Testing

#### **Drawbacks of Automation Testing**

- 1) It is expensive
- 2) We cannot automate all areas.

- 3) Lack of expertisation.
- 4) It has some limitations (It cannot test every thing)

#### **Which Software Testing should be automated?**

Tests that need to be executed on every build of the application (Sanity Testing)

Tests that use multiple data values (Retesting / Data Drives Testing)

Tests that require data from application intimates (G.U.I. Attributes) Load and Stress Testing

#### **Which Software Testing should not be automated?**

Usability Testing One time testing

Quick look Tests or A.S.A.P (As soon as possible) Testing Ad-hoc testing / Random Testing

Customers requirement are frequently changing.

---

#### **Types of Test tool:**

---

Business:

a) Vendor tools

Ex: HP- WinRunner, LoadRunner, QTP, QC  
IBM-Rational Robot, RFT, RPT, QA Director  
Borland-SilkTest, Silk Performer etc..

b) Open Source Tools:

Ex: Selenium, Jmeter, QAWebLoad, Bugzilla etc...

c) In-house tools:

Technical:

a) Functional & Regression Test Tools:

Ex: WinRunner, QTP, Rational Robot, RFT, SilkTest, Selenium etc..

b) Performance/load/stress test tools

Ex: LoadRunner, RPT, Silk Performer, Jmeter, QAWebLoad etc...

c) Test Management Tools:

Ex: QC, QA Director Etc...

d) Defect Management tools

e) Unit Test tools (Ex: JUnit)

## **QTP Basic Features**

- QTP was Launched in 2002 (Nov). By Mercury Interactive. Later taken over by HP, in 2007.
- QTP is an **Object Based** Testing Tool.

*(Object based means; QTP Follows Test Object Model for Performing Testing operations. Based on either Test Objects or Automation Objects or Utility Objects only, we can automate test operations)*

*(LoadRunner is a Protocol based Test Tool)*

- QTP is for Functional and Regression Testing.

*(Basically QTP is a Functional and Regression Test tool but we can use it for little bit Compatibility Testing and Performance Testing)*

- It follows **Keyword Driven** Approach.

*(Keyword Driven Approach means; Keywords, example: Commands, functions, methods, statements etc.. are used for creating Test scripts)*

- It supports Windows Operating Environment only.

*(QTP basically depends on User Interface to automate test operations and it is not depends on server side interactions, where as LoadRunner depends on server side Operations, so HP developed LoadRunner for Windows as well as UNIX versions)*

- It supports GUI based (Graphical user interface) and Web based Applications Automation, does not support CUI (Command user interface) Applications.
- It has multilingual support. (It supports English, Chinese, Japanese, Korean, etc languages with respect to license.)
- It has adapted Microsoft Visual Basic Scripting edition for programming and Excel like spread sheet for Data driven testing.

#### **Tools support for QTP:**

- VB Script for Scripting (for implementing programming logic)
  - VB Script engine integrated in QTP, so apart from scripting features we can use File system, Excel, word, Adodb etc. Object models from QTP.
  - SQL (Structured query language) engine integrated in QTP, so we can use SQL Statements in QTP to perform Database Operations
  - Excel like spread sheet integrated in QTP for performing Data driven Testing
  - We can install MS Script Debugger with QTP for Debugging Test scripts.
  - QTP uses XML format for Test results, Environment variables and for exporting object repositories.
- 
- QTP can be integrated with other Tools like WinRunner and Quality Center.
  - QTP can be used for user interface (UI) test case automation and some limited (Non UI) Test case automation like File system operations and Data base operations.
  - It was derived from Astra QuickTest (mother tool of QTP).
  - It supports IE (Internet Explorer) only for recording tests, for execution it supports other browsers like Mozilla and AOL, opera Etc apart from IE.
  - QTP is a Desktop Application (I-Tier/Stand alone).
  - QTP developed in .NET Technology.

• Since it is Stand-alone Application, It doesn't have Database, It stores resources as files (Internal and External). We can't share qtp services from one machine to another, but we can share resources(Ex: Object Repository files, Function Libraries, Environment variable files, Recovery scenario files etc...)

**QTP has two types of License;**

- **Seat** or Node locked License,
- **Concurrent** or float license.
- We have to use Seat license from fixed system/machine and Concurrent License, we can use from any system but one user at a time. If we want multiple concurrent licenses that we can purchase.
- QTP has UNI code support.

#### **Version History of QuickTest Professional**

- Astra QuickTest (Mother tool of QTP) 1.0 to 5.0

\* QTP derived from Astra QuickTest and influenced by WinRunner.

Quick Test Professional

- 5.6 – Nov 2002
- 6.5 – 2003
- 8.0 – 2004
- 8.2, 8.3 –2005
- 9.0,9.1-2006
- 9.2- 2007 Mercury Interactive
- 9.5 –2008 H.P
- 10.00 –2009 January
- **11.00 –2010\*\*** (It is Latest Version)

#### **1) Test Pane:**

It is programming interface of QTP, used for creating, editing, viewing and deleting test scripts.

**Test:** It is a set of statements or set of one or more actions. (It is a local concept)

**Action:** It is a set of Statements. (It is a local concept)

**Statement:** A minimal executable unit. Statements may have keywords. (Keyword such as function, method, statement etc). It is a global concept.

**Test pane has 2 views.**

Expert view- Test in VB script format.

Key word view- Test in Icon based GUI format.

**Note1:** Here in test pane Test is same but views are different, if you perform any modifications in one view those automatically reflects in another view.

**Note 2:** Technical users use expert view and business user use key word view.

**Note 3:** User can customize Test pane view options.

Navigation: Tools Menu> view options> we can use font size and colors etc.

## **2) Active Screen**

It has an optional element of QTP, it takes snap shots of every statement that user performed action or operation on AUT.

It can be used for understanding and editing the test easily.

**Navigation:** View menu>active screen (for launching and closing)

**Configuring Active Screen:** Tools > Options > Active screen tab > Increase / Decrease capture level, Apply and OK.

Note 1: Generally novice testers use this feature but it occupies lot of memory space on QTP.

## **3) Data Table**

It has an integrated spread sheet (Excel like), used for Data driven testing.

Navigation: View menu>Data table (for launching and closing)

Data driven Testing: Testing the same task(S) or same operation with multiple sets of test data.

It can be used in two ways.

1. Enter test data directly into Data Table and use.

2. Importing data from external files (Flat files, excel sheets, etc) and Data bases (MS access, Sql Server, oracle, etc.)

It has two types of sheets.

- 1). Global Sheet- for entire Test /used for all actions in the Test.

- 2). Action Sheets- for specific Action only.

Data table available in 2 ways:

- 1). Design time data table. (Attached with every test)

- 2). Run time data table. (Available in result window)

Run time is a carbon copy of design time data table.

QTP has Data table methods for performing operations on data table.

#### **4) Debug Viewer**

It is used for locating and rectifying or fixing errors.

It can be used for debugging the tests by step-by-step execution.

We use Step into, Step over and Step out commands for debugging. (debug commands)

Navigation: View menu>debug viewer (for launching and closing)



Through Debug viewer user can watch variables and change values of variables temporarily.

## **5) Missing Resources**

It can be used for showing missing resources that attached to current test such as (Recovery Scenarios, Library files etc).

Navigation: View menu > missing resources (for launching and closing)

## **6) Information**

This pane shows syntax errors automatically during saving the test.

Navigation: view menu > information

## **7) QTP Commands**

QTP Commands are available in 3 ways.

- 1). Menu options
- 2). Tool Bar options
- 3). Short cut keys (for Some important operations only)

## **QTP Tool Window Menus**

File menu: Through file menu user can create, save tests, open existing tests, export tests in zip format.

Edit Menu: It provides editing options and renaming, deleting and splitting actions.

View menu: Through this menu we can launch and close, active screen, Data Table, Debug viewer, information, missing resources etc.

Insert Menu: Through this menu user can inserting check points, out put values, synchronizing points.

In this menu step generator available, using this user can generate recordable and non-recordable scripts.

Through insert menu user can insert VB Script conditional and loop statements and transaction points (Start and End).

Through insert menu user can create new actions, call existing actions and copy existing actions.

Automation Menu: This menu provides Record, Run options and Run setting options

Through this menu we can start normal recording, analog recording and Low level recording.

Through this menu we can stop recoding, running and also we run tests.

Resources Menu: This menu provides object repository and recovery scenarios options.

Through this menu we can create /modify/delete objects information and we can associate repositories.

Through this menu we can create, modify and delete recovery scenarios.

Debug Menu:

This menu provides debug commands for step by step execution.

Through this menu we can insert/remove/break points.

Tools Menu: This menu provides Tools settings option, view options and object identification configuration.

Through this menu we can set tool options as well as test pane view options.

In this menu object spy option available, through this we can get object's information.(Properties and values)

In this menu Virtual object option available; through this option we can create virtual objects.

Window Menu: This menu provides QTP tool window style settings.

Help Menu: This menu provides QTP help as well as VB Script help.

Through this menu we can contact technical support people and we can send feedback.

Through this menu we can check for updates and download or install directly.

- [QTP Guide](#)
- [QTP Online Training](#)
- [QTP Scripting](#)
- [QTP Training in Hyderabad](#)
- [SQL](#)
- [VB Script](#)

## Apr 5, 2010

### QTP Commands

#### QTP Commands

**QTP Commands are available in 3 ways.**

- 1.Menu options
- 2.Tool Bar options
- 3.Short cut keys (for Some important operations only)

**File menu:** Through file menu user can create, save tests, open existing tests, export tests in zip format.

<b>Command</b>	<b>Function</b>
<b>New &gt; Test</b>	Creates a new test.
<b>New &gt; Business Component</b>	Creates a new business component.
<b>New &gt; Scripted Component</b>	Creates a new scripted component.
<b>New &gt; Application Area</b>	Creates a new application area.
<b>New &gt; Function Library</b>	Creates a new function library.
<b>Open &gt; Test</b>	Opens an existing test.
<b>Open &gt; Business/Scripted Component</b>	Opens an existing business or scripted component.
<b>Open &gt; Application Area</b>	Opens an existing application area.
<b>Open &gt; Function Library</b>	Opens an existing function library.
<b>Close</b>	Closes the active function library.
<b>Close All Function Libraries</b>	Closes all open function libraries.
<b>Quality Center Connection</b>	Opens the Quality Center Connection dialog box, enabling you to connect to a Quality Center project.
<b>Quality Center Version Control</b>	Provides a sub-menu of options for managing versions of QuickTest assets in Quality Center. The sub-menu is available only when you are connected to version-control enabled Quality Center project.
<b>Save</b>	Saves the active document.
<b>Save As</b>	Opens the relevant Save dialog box so you can save the open document.
<b>Save Test with Resources</b>	Saves a standalone copy of the current test together with its resource files.
<b>Save All</b>	Saves all open documents.
<b>Enable Editing</b>	Makes read-only function libraries editable.
<b>Export Test to Zip File</b>	Creates a zip file of the active document.
<b>Import Test from Zip File</b>	Imports a document from a zip file.
<b>Convert to Scripted Component</b>	Converts a business component to a scripted component.
<b>Print</b>	Prints the active document.
<b>Print Preview</b>	Displays the Keyword View as it will look

	when printed and enables you to modify the page setup.
<b>Settings</b>	Opens the Settings dialog box, enabling you to define settings for the open document. (Not relevant for function libraries)
<b>Process Guidance Management</b>	Opens the Process Guidance Management dialog box, enabling you to manage the list of processes that are available in QuickTest.
<b>Associate Library</b> <b>‘&lt;Function Library Name&gt;’ with ‘&lt;Document Name&gt;’</b>	Associates the active function library with the open document. (Available only from function libraries)
<b>Recent Files</b>	Lists the recently viewed files.
<b>Exit</b>	Closes the QuickTest session.

**Edit Menu:** [It provides editing options and renaming, deleting and splitting actions.](#)

<b>Command</b>	<b>Function</b>
<b>Undo</b>	Reverses the last command or deletes the last entry you typed.
<b>Redo</b>	Reverses the most recent operation of the Undo command.
<b>Cut</b>	Removes the selection from your document.
<b>Copy</b>	Copies the selection from your document.
<b>Paste</b>	Pastes the selection to your document.
<b>Delete</b>	Deletes the selection from your document.
<b>Copy Documentation to Clipboard</b>	Copies the content of the <b>Documentation</b> column of the Keyword View, enabling you to paste it in an external application.
<b>Action &gt; Split Action</b>	Separates an action into two sibling actions or into parent-child nested actions.
<b>Action &gt; Rename Action</b>	Changes the name of an action.
<b>Action &gt; Delete Action</b>	Enables you to remove the selected call to the action, or delete the action and its calls from the active test.

<b>Action &gt; Action Properties</b>	Enables you to specify options, parameters, and associated object repositories for a stored action.
<b>Action &gt; Action Call Properties</b>	Enables you to specify the number of run iterations according to the number of rows in the Data Table, and to define the values of input parameters and the storage location of output parameters.
<b>Step Properties &gt; Comment Properties</b>	Opens the Comment Properties dialog box for a comment step. Available only when the selected step is a comment.
<b>Step Properties &gt; Object Properties</b>	Opens the Object Properties dialog box for a selected object. Available only when the selected step contains a test object.
<b>Step Properties &gt; Checkpoint Properties</b>	Opens the relevant Checkpoint Properties dialog box for a selected object. Available only when the selected step is a checkpoint step.
<b>Step Properties &gt; Output Value Properties</b>	Opens the relevant Output Value Properties dialog box for a selected object. Available only when the selected step is an output value step.
<b>Step Properties &gt; Report Properties</b>	Displays the Report Properties dialog box for a report step. Available only when the selected step is a <b>Reporter.ReportEvent</b> step.
<b>Find</b>	Searches for a specified string.
<b>Replace</b>	Searches and replaces a specified string.
<b>Go To</b>	Moves the cursor to a particular line in the test.
<b>Bookmarks</b>	Creates bookmarks in your script for easy navigation.
<b>Advanced &gt; Comment Block</b>	Comments out the current row, or selected rows.
<b>Advanced &gt; Uncomment Block</b>	Removes the comment formatting from the current or selected rows.
<b>Advanced &gt; Indent</b>	Indents the step according to the tab spacing defined in the Editor Options dialog box.
<b>Advanced &gt; Outdent</b>	Outdents the step (reduces the indentation) according to the tab spacing defined in the Editor Options dialog box.
<b>Advanced &gt; Go to Function Definition</b>	Navigates to the definition of the selected function.
<b>Advanced &gt; Complete Word</b>	Completes the word when you type the beginning of a VBScript method or object.

<b>Advanced &gt; Argument Info</b>	Displays the syntax of a method.
<b>Advanced &gt; Apply “With” to Script</b>	Generates <b>With</b> statements for the action displayed in the Expert View, and enables IntelliSense within <b>With</b> statements.
<b>Advanced &gt; Remove “With” Statements</b>	Converts any <b>With</b> statements in the action displayed in the Expert View to regular (single-line) VBScript statements.
<b>Optional Step</b>	Inserts an optional step (a step that is not required to successfully complete a run session).

**View menu:** Through this menu we can launch and close, active screen, Data Table, Debug viewer, information, missing resources etc.

**Insert Menu:** Through this menu user can inserting check points, out put values, synchronizing points.

In this menu step generator available, using this user can generate recordable and non-recordable scripts.

Through insert menu user can insert VB Script conditional and loop statements and transaction points (Start and End).

Through insert menu user can create new actions, call existing actions and copy existing actions.

#### **Automation Menu:**

This menu provides Record, Run options and Run setting options

Through this menu we can start normal recording, analog recording and Low level recording.

Through this menu we can stop recoding, running and also we run tests.

<b>Command</b>	<b>Function</b>
<b>Record</b>	Starts a recording session.
<b>Run</b>	Starts a run session from the beginning or from the line at which the session was paused.

<b>Stop</b>	Stops the recording or run session.
<b>Run Current Action</b>	Runs only the active action.
<b>Run from Step</b>	Starts a run session from the selected step.
<b>Maintenance Run Mode</b>	Starts a run session during which the Maintenance Run Mode wizard opens for steps that failed because an object was not found in the application (if applicable).
<b>Update Run Mode</b>	Starts a run session to update test object descriptions and other options (if applicable).
<b>Analog Recording</b>	Starts recording in <b>Analog Recording</b> mode.
<b>Low Level Recording</b>	Starts recording in <b>Low Level Recording</b> mode.
<b>Record and Run Settings</b>	Opens the Record and Run Settings dialog box, enabling you to define browser preferences for recording and running your test.
<b>Process Guidance List</b>	Lists the processes that are available for the current document type and for the currently loaded QuickTest add-ins, enabling you to open them.
<b>Results</b>	Enables you to view results for a test run session.

#### Resources Menu:

This menu provides object repository and recovery scenarios options.

Through this menu we can create /modify/delete objects information and we can associate repositories.

Through this menu we can create, modify and delete recovery scenarios.

<b>Command</b>	<b>Function</b>
<b>Object Repository</b>	Opens the Object Repository window, which displays a tree containing all objects in the current test or component.
<b>Object Repository Manager</b>	Opens the Object Repository Manager dialog box, enabling you to open and modify multiple shared object repositories.
<b>Associate Repositories</b>	Opens the Associate Repositories dialog box,



	enabling you to manage the object repository associations for the test.
<b>Map Repository Parameters</b>	Opens the Map Repository Parameters dialog box, enabling you to map repository parameters, as needed.
<b>Recovery Scenario Manager</b>	Opens the Recovery Scenario Manager dialog box.
<b>Associated Function Libraries</b>	Lists the function libraries associated with the active document, enabling you to open them.

#### Debug Menu:

This menu provides debug commands for step by step execution.

Through this menu we can insert/remove/break points.

#### Tools Menu:

This menu provides Tools settings option, view options and object identification configuration.

Through this menu we can set tool options as well as test pane view options.

In this menu object spy option available, through this we can get object's information.(Properties and values)

In this menu Virtual object option available; through this option we can create virtual objects.

Command	Function
<b>Options</b>	Opens the Options dialog box, enabling you to modify global testing options.
<b>View Options</b>	Opens the Editor Options dialog box, enabling you to customize how tests and function libraries are displayed in the Expert View and Function Library windows.
<b>Check Syntax</b>	Checks the syntax of the active document.
<b>Object Identification</b>	Opens the Object Identification dialog box, enabling you to specify how QuickTest identifies a particular test object.
<b>Object Spy</b>	Opens the Object Spy dialog box, enabling you to view the native properties and operations of any object in an open application, as well as the test object hierarchy, identification properties, and operations of the test object that QuickTest uses to represent that object.
<b>Web Event Recording</b>	Opens the Web Event Recording Configuration dialog box,

<b>Configuration</b>	enabling you to specify a recording configuration level. (Relevant for tests only)
<b>Data Driver</b>	Opens the Data Driver dialog box, which displays the default Constants list for the action. (Relevant for tests only)
<b>Change Active Screen</b>	Replaces the previously recorded Active Screen with the selected Active Screen. (Relevant for tests only)
<b>Virtual Objects &gt; New Virtual Object</b>	Opens the Virtual Object Wizard, enabling you to teach QuickTest to recognize an area of your application as a standard test object.
<b>Virtual Objects &gt; Virtual Object Manager</b>	Opens the Virtual object Manager, enabling you to manage all of the virtual object collections defined on your computer.
<b>Customize</b>	Opens the Customize dialog box, which enables you to customize toolbars and menus, and create new menus.

## QTP Test Process

### 7 Stages of QTP Testing Process

#### 1) Planning

- o Analyzing the AUT
- o Automation Test Plan Generation
- o Automation Framework Implementation
- o Generating/Selecting Test cases for Automation
- o Collecting Test Data
- o QTP Tool Settings Configuration

#### 2) Generating Tests

- o Recording
- o Keyword driven methodology

- o Descriptive Programming

### **3) Enhancing Tests**

- o Inserting Checkpoints
- o Inserting Output values
- o Adding Comments
- o Synchronization
- o Parameterization
- o Inserting Flow Control Statements
- o Calling User defined functions and/or Reusable Actions
- o Generating Steps through Step Generator
- o Inserting Transaction Points
- o Regular Expressions

### **4) Debugging Tests**

- o Debug Commands & Break Points
- o Step by step execution
- o Watching Variables
- o Changing values of variables

### **5) Running Tests**

- o Normal Execution
- o Batch Execution
- o Through AOM Scripting
- o Tests Running through framework

- o Scheduled Execution

## **6) Analyzing Results**

- o QTP Result window
- o Defining our own Results
- o Exporting Results
- o Deleting Results

## **7) Reporting Defects**

- o Manual Defect Reporting
  - o Tool based Defect Reporting
  - o Working with Quality Center
- 

### **Types of Statements in QTP Test / Test Script**

#### **i) Declarations** (Variables and constants)

Dim x, y, z  
Const City, Price

#### **ii) Object calls**

Ex1: Dialog("Login").WinEdit("Agent Name:").Set "Naseer"  
Ex2: Browser("Google").Page("Google").Link("Gmail").Click

#### **iii) Comments**

#### **iv) Flow Control Statements** (Conditional & Loop)

Ex:) If Total=Tickets\*Price Then  
Msgbox "Test Passed"  
Else  
Msgbox "Test Failed"  
End If

#### **v) Function / Action calls**

Ex: Call Login("Naseer","mercury")  
vi) Utility Statements  
Ex1: SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"  
\*\*It launches the Application  
Ex2:  
SystemUtil.Run "C:\Program Files\Internet Explorer\IEXPLORE.EXE","http://www.icicibank.com/"

#### **vii) VB script other statements**

Examples:  
Option Explicit  
Wait (14)

### **Recording and Running**

#### **a) Test Recording Process**

It is a Process of Recording user operations on AUT (Application Under Test). During Recording QTP Creates steps in Keyword view, and generates them in a script in the Expert view. Simultaneously it adds Objects information into Object Repository.

#### **b) Running /Execution Process**

During Running QTP reads statements one by one and gets Object Information from the Object Repository, based on that Information performs operations on AUT.

#### **c) Recording Modes**

QTP has 3 Recording Modes to generate Tests / Test Scripts

#### **i) Normal Recording**

It records User Mouse and Keyboard operations on AUT with respect to objects, but unable to record continuous mouse operations like Digital Signatures, graphs, paints etc.

During recording QTP generates VbScript statements in Test Pane, Simultaneously it stores objects information into object repository.

**Navigation:** Automation>Record

Or

Select Record option on automation toolbar

Or

Use short cut key (F3)

**Steps for preparing a Test** (through Recording):

1.Put AUT in base state

2.Select Record Option

3.It shows Record and Run Settings, Select type of Environment (Windows or Web)

4.Select Record Option

(It shows two Options :

1.Record and Run Test on any open window based applications

2.Record and Run only on)

If we select first option it records on any opened application on Desktop.

If we select Second option, it asks for the path of the AUT, After Providing the path it records only on that particular application.)

5.Click OK

6.Perform actions on AUT

7.Stop recording.

8.Save the Test

## **ii) Analog Recording:**

It records the exact mouse and keyboard operations. We can use this mode for recording continuous mouse operations. It is not useful for recording normal operations why because it does not generate steps for each operation, generates total user actions in a Track File. The Track file is not editable.

### **Navigation:**

1.Keep tool under recording mode

2.Automation >Analog Recording

OR

Use Short cut Key (Shift + ALT+F3)

**Steps for preparing a Test** (through Analog Recording):

1.Launch AUT (or we can launch AUT through QTP)

2.Select Record option

3.Automation>Analog Recording

4.Analog Recording Settings Dialog box opens

(In this Dialog box two options available.

1.Record relative to the screen

2.Record relative to the following window)

(If we select first option QTP records User operations with respect to Desktop co-ordinates.

If we select Second option, we have to show the window (AUT), after showing the Window it records with respect to that window co-ordinates.)

5.Select any one option in the dialog box and click Start Analog record.

6.It records User actions

7.Stop Recording

### **iii) Low Level Recording**

It records some operations on Non-supported environments apart from Normal operations.

This mode records at the object level and records all run time objects as window or winobject Test objects.

Use Low Level Recording for recording in an environment not recognized by QTP.

### **Navigation:**

- 1.Keep tool under recording mode
- 2.Automation >Low Level Recording

**Steps for preparing a Test** (through Low Level Recording):

- 1)Launch AUT (or we can launch AUT through QTP)
- 2)Select Record option
- 3)Automation> Low Level Recording
- 4)Perform options on AUT
- 5)Stop Recording
- 6)Save the Test

**d) Disadvantages of Recording**

- It occupies a lot of memory space(due to duplicate objects), So QTP performance will be reduced
- No Centralized Maintenance mechanism , So Modifications are very difficult
- User may not have command on the Recorded script, So locating errors is difficult
- Recorded scripts are QTP internal files, they may corrupt.

**e) Advantages of Recording/ Where Applicable**

- It is used for Analyzing the AUT in the initial stage to find out weather the QTP tool is Recognizing all of our Application Objects or not
- It is easy to create Tests / Test Scripts
- It is used for frequently changing UI (User Interface)
- It takes less time to create Tests



# Object Repository

## **Object Repository:**

It is a storage place of QTP where objects information can be stored and it also acts as interface between the Test script and the AUT in order to identify the objects during execution.

## **Object:**

Object is something, which has structure and properties.

## **Software objects:**

We call Windows, Web Pages, Buttons, Edit boxes, Check boxes etc.. as software objects.

## **Types of Object in QTP:**

There are four types of object available in QTP.

1. Run time objects
2. Test objects
3. Utility objects
4. Automation objects/User defined objects.

**Run time objects:** The objects present in the AUT. Ex: Buttons, Links, etc...

**Test Objects:** References of Run time objects. Ex: WinEdit, WinButton, WebButton, Link, etc...

## **Note:**

Test objects names vary from one environment to another.

## **Utility objects**

They are QTP reserved objects used for Testing and Result reporting.

Ex:

1. SystemUtil for launching/closing the application.
2. Reporter for defining results.
3. Services for inserting transaction points
4. Environment for using environment variables

#### **Automation objects/User defined objects**

User can create objects; those can be used for performing specific operations. Ex: Creating objects in filesystemobject class, adodb.connection class, dictionary object class, Excel. Application class etc.

#### **There are two types of repository available in QTP.**

1. Local repository (.MTR extension)
2. Shared repository(.TSR extension)

#### **Local Repository:**

QTP creates a Local Repository for every Action automatically during Recording. That cannot be shared among tests.

User can add some more objects to Local repository

User can perform rename, delete operations on local repository. As it is QTP internal file user no need to save modifications.

#### **Shared Repository:**

User (Test Engineer) creates the Shared Repository by adding objects. That can be shared among number of tests.

Using this method user can perform modifications on objects easily.

## **Operations on Object Repository**

### **Adding objects**

a. Local Repository:

Navigation: open Local Repository (Resource Menu > Object Repository)

objects> Add objects to Local> Show the Object>Click Ok

(No need to save separately, it saves automatically)

### **B. Shared Repository:**

Navigation: Resource menu>object repository manager>object> Add objects>show the window/object>click ok

o Selected object only

o Default object types

o All objects types

o Selected object types

(If we select first option it stores Selected objects only, if we select second option it stores all default objects, if we select third option it stores All objects including static objects, if we select fourth option, we have to select object classes then it stores that class objects, we can select more than one class also.)

### **Renaming Objects**

a. Local Repository:

Resources > object repository > select object and right click > choose rename option > modify the name > release the mouse. > close repository

b. Shared Repository:

Resources > object repository manager > file > open > browse path of the repository file > file > enable editing > select object & Right click > choose rename option > modify the name > release mouse > save the file & close repository manager.

### **Deleting Objects**

a. Local Repository:

Resources > object repository > select object and right click > choose delete option > confirm deletion > . > close repository

b. Shared Repository:

Resources > object repository manager > file > open > browse path of the repository file > file > enable editing > select object & Right click > choose delete option > confirm the deletion > save the file & close repository manager.

### **Associating Object Repositories to an Action/Test**

Resources > Associate repositories > click add icon (+) > browse path of the repository > Associate with an action > click ok

### **Merging Repositories**

Resources > object repository manager > tools > object repository merge tool > browse path of the first repository > browse path of the second repository > click ok > click close > save the merged repository > close the repository manager.

**Note:** After merging also, source repository files will be available, if we do not want source files we can delete.

## Defining New Test Objects

Navigation: Object>Define new test object >Select Environment>Select Class of the object>Enter name of the object>Click Add>click close>Select object >select property name>enter value (like this select one by one properties and enter values)>save

**Note:** This feature can be used for preparing tests before the AUT is ready.

## Spying Objects

For getting objects information, (Test objects names, property & Values) QTP is providing a feature called Object Spy, using this we can get objects information.

Navigation>Tools>object spy>take hand icon & Show the object>get information (Object Spy shows the specific objects all available properties with their values)

**Note:** As Object spy is an important feature, it can be available in 3 Areas.(1. In tools Menu 2. In local repository 3. In Repository manager)

## View Options

Locate in Repository

This feature can be used for identifying objects from application to repository.

Navigation: View>Locate in repository >show the object>click Ok>it locates specified object in the repository.

Highlight in Application

This feature can be used for identifying objects from repository to application.

Navigation: Select object in the repository >view>highlight in application>it highlights specified objects in the application.

### **Exporting Repository to an XML file**

We can export our repository file to an xml file.

Navigation: File > export Test objects to XML Enter the file name and Save with xml extension.

### **Importing Repository from XML file.**

We can import XML file to our repository.

Navigation: File>import from XML >browse path of the XML file >it loads objects.

Note: The purpose of this feature is Editing objects information outside of the QTP

Navigation: Resource menu>object repository (Short cut key Ctrl+R)

## **QTP Methods / Operations**

### **1) Activate Method**

It activates dialog boxes/Windows

Syntax: Object hierarchy. Activate

Ex:

Dialog("Login").Activate

Window("Flight Reservation").Activate

### **2) Click Method**

It clicks on an object (Buttons)

Syntax: Object hierarchy. Click

Ex:

Dialog("Login").Winbutton("ok").click

### **3) Close Method**

It closes the window, Dialog box, Browser window etc.

Syntax: Object hierarchy. Close

Ex:

Window("Flight Reservation").Close

Browser("Browser").Close

### **4) Dblclick**

It Double clicks the object.

### **5) Set Method**

It can be used in three ways.

a. For setting the value of an edit box

Syntax: Object Hierarchy. SET "Value"

Ex: Dialog ("Login").WinEdit("Agent Name").Set "asdf"

b. Selecting a Radio Button

Syntax: Object Hierarchy. Set

Ex: Window("Flight Reservation").WinRadioButton("Business").Set

c. Selecting/Deselecting check boxes

Syntax: object Hierarchy. Set "ON/off"

Ex: Window ("Flight Reservation"). Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

#### 6) **Select** Method:

It is used for selecting an item from a combo box or list box.

Syntax: Object hierarchy.select "item"

Ex:

```
Window("Flight Reservation").WinComboBox("Fly From:").Select "London"
```

```
Window("Flight Reservation").Dialog("Flights Table").WinList("From").Select "12572"
```

#### 7) **GetVisibleText**: It returns Text from the specified area.

Syntax: Variable =object hierarchy.getvisibletext

Ex:

```
x=Window("Flight Reservation").WinComboBox("Fly From:").GetVisibleText
```

```
msgbox x
```

#### 8) **GetROProperty**:

It returns current object property value. It can be used for getting any object's, any property value.

Syntax: Variable = object hierarchy.getroproperty ("property name")

Ex: x=Window("Flight Reservation").WinEdit("Tickets:").GetROProperty ("width")

```
msgbox x
```

#### 9) **Navigate**Method:

It opens a specified URL in the Browser.

Syntax: object hierarchy.navigate "URL"

Ex: Browser("Yahoo!").Navigate <http://www.google.co.in/>

#### 10) **GetItemsCount**: It returns number of items in a combobox.

Syntax: Variable=object hierarchy.getitemscount

Ex: x=Window("Flight Reservation").WinComboBox("Fly From:").GetItemsCount



msgbox x

**11) Getcontent:** It returns all items from a combobox.

Syntax: variable=object hierarchy.GetContent

Ex: x=Window("Flight Reservation").WinComboBox("Fly From:").GetContent

msgbox x

**12) Exist property:** It checks whether the object is available or not.

Syntax: Object hierarchy.exist(time in seconds)

Ex: Window("Flight Reservation").Exist(5)

## Object Identification Configuration

### Overview:

Generally for every object 20-25 properties information available, qtp recognizes object using 2 Or 3 important properties.

Qtp has default object identification configuration for every environment, if we feel that config is not sufficient for recognizing objects in our application, we can configure some more

### Object Identification Types

#### a) Normal identification

- 1) Mandatory properties
- 2) Assistive properties

#### b) Smart identification

- 1) base filter properties
- 2) optional filter properties

#### c) Ordinal identifier

- 1) location
- 2) index
- 3) creation time(only for web)

### QTP learns information in the following in case of normal identification:

First of all the qtp learns all the mandatory properties at a time and thinks whether these properties sufficient to identify the object uniquely. if it feels sufficient then it stops learning otherwise, It learns first assistive property and once again stops and thinks, like this qtp learns one by one. At the end of assistive properties list also if it feels not satisfied and it finally goes to Ordinal Identifier.

### QTP learns information in the following in case of Smart Identification:

Smart identification is an optional feature, if we feel normal identification is not sufficient for any object, and then we configure Smart Identification for that object, in order to avoid Ordinal Identifier.

After normal identification if qtp feels not satisfied then it goes to smart identification. in smart identification 2 types of properties available, first qtp learns all base filter properties at a time and thinks whether these properties are sufficient for identifying the object uniquely. If it feels sufficient, then it stops learning otherwise it goes Optional Filter Properties and learns one by one. Still it feels not satisfied finally it goes to Ordinal Identifier.

#### **Ordinal identifiers:**

There are 3 types of ordinal identifiers available

- 1) Location: is based on object location in the AUT ,location starts from zero.
- 2) index: it is based on sequence of the programs, index starts from zero
- 3) Creation time: it is based on loading time of the web objects. qtp generates 0,1,2 like numbers.

#### **Tool Settings Globalization:**

As QTP is a I-tier(Stand-alone) application,making Tool settings globally is not possible.

For making tool settings global, QTP is providing a special feature called "Generate Script".

#### **STEPS:**

- 1) Settings available in 3 areas.
  - a) File->Settings
  - b) Tools->Options
  - c) Tools->Object Identification
- 2) Perform required settings and generate Scripts
- 3) Share Script files to team members and ask them to execute those scripts.

#### **QTP Data Table Methods**

##### **(1) Add Sheet:**

We can use this method to "add" one new sheet to the run time data table.

Syntax: DataTable.AddSheet "sheet name"

Ex: DataTable.AddSheet "Naseer"

##### **(2) Delete Sheet:**

We can use this method to "delete" one specified sheet from the Run Time Data table.

Syntax: datatable.DeleteSheet (Sheet\_ID)

Ex: datatable.DeleteSheet (3)

##### **(3) GetSheetCount**

We can use this method to count number of sheets in the run time data table.

Syntax:

datatable.GetSheetCount

msgbox datatable.GetSheetCount

##### **(4) GetRowCount**

We can use this method to count number of rows in the 1st sheet (longest column) of the Run time data table.

Syntax: datatable.GetRowCount

Ex: msgbox datatable.GetRowCount

#### (5) **GetSheet**

We can use this method to return a specified sheet from the Run Time data table.

Syntax: datatable.GetSheet(SheetID)

Ex: msgbox datatable. GetSheet(1).GetRowCount

#### (6) **Value**

We can use this method to set or get value of cell in the specified parameter and the current row of the Run time data table.

##### **To set data**

Syntax: datatable.Value(Parameter\_Name, Sheet\_Name) = Value / variable Or

datatable(Parameter\_Name, Sheet\_Name) = Value / variable

##### **To get data**

Syntax: Variable = datatable.Value(Parameter\_Name, Sheet\_Name) Or

Variable = datatable(Parameter\_Name, Sheet\_Name)

Ex:

Option explicit Dim a, b, c

a=datatable.Value (1,1) b=datatable.Value (2,1) c=cint(a)+cint(b) datatable.Value (3,1) = c

Note: Default property of Datatable is value

#### (7) **SetCurrentRow**

We can use this method to take a specified row as current row in the Run Time Datatable (By default it is 1st Row of 1st Sheet)

Syntax: datatable.SetCurrentRow(Row\_Number)

g="Naseer"

datatable.SetCurrentRow (3) datatable.Value (1,1) = g

#### (8) **SetNextRow**

We can use this method to take the row after the current Row as New Current Row in the Run time data table.

Syntax: datatable.SetNextRow

Ex: g="Naseer"

datatable.SetCurrentRow (3) datatable.SetNextRow datatable.Value (1,1) = g

#### (9) **SetPrevRow**

We can use this method to take the row before the current Row as New Current Row in the Run time data table.

Syntax: datatable.SetPrevRow

Ex: g="Naseer"

datatable.SetCurrentRow (3) datatable.SetPrevRow datatable.Value (1,1) = s

#### (10) **Import**

We can use this method to import Microsoft Excel File to the Runtime Data Table (Including all sheets)

Syntax: datatable.Import "Path of File"

Ex: datatable.Import "F:\Inputdata.xls"

#### 11) **ImportSheet**

We can use this method to import a specified sheet of Microsoft Excel Sheet to the Runtime Data table.

Syntax: datatable.ImportSheet "Path of File", "Source Sheet", "Destination Sheet"

Ex: datatable.ImportSheet "E:\Naseer.xls",3,1

#### (12) **Export**

We can use this method to export a copy of Run Time Data table to another location (Including all sheets)

Syntax: datatable.Export "Path of File"

Ex: datatable.Export "F:\Naseer.xls"

#### 13) **ExportSheet**

We can use this method to export a copy specified sheet of Run Time Data table to the existing or new Excel File.

Syntax: datatable.ExportSheet "Path of File", "Sheet Name / Source Sheet"

Ex: datatable.ExportSheet "F:\Naseer.xls", 2

\*\*\*\*\*

#### **Data Driven Testing for Login Operation using Data Table methods**

Datatable.AddSheet "Login"

Datatable.ImportSheet "C:\Documents and Settings\gcr\Desktop\Naseer.xls",1,3

Rows\_count=Datatable.GetSheet(3).GetRowCount

For i= 1 to Rows\_count

Datatable.SetCurrentRow(i)

SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\HP\QuickTest Professional\samples\flight\app\", "open"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set Datatable("Agent",3)

Dialog("Login").WinEdit("Password:").Set Datatable("Pwd",3)

Dialog("Login").WinButton("OK").Click

If window("Flight Reservation").exist(10) Then

Login="Login Operation Successful"

```
Datatable("Result",3)=Login
Reporter.ReportEvent micPass,"res","Passed"
else
Reporter.ReportEvent micFail,"res","Failed"
Login="Login Operation Failed"
Datatable("Result",3)=Login
```

```
Dialog("Login").Dialog("Flight Reservations").WinButton("OK").Click
Dialog("Login").WinButton("Cancel").Click
End If
```

```
If Window("Flight Reservation").Exist(3)Then
Window("Flight Reservation").Close
End if
Next
```

### **Descriptive Programming**

Entering / Providing objects information directly into the test script is called Descriptive Programming.

In this method of script creation, we no need to have Object Repositories.

#### **Advantages:**

- a) Descriptive Programming based Test scripts are faster in execution than Repository based Test scripts.
- b) Scripts are portable (we can run these scripts from any machine easily)
- c) Maintenance is easy (less amount of resources)
- d) We can start Test Execution process even though Application is not ready.

**Descriptive programming is basically 2 types.**

1. Static Programming
2. Dynamic Programming

### **Static Programming**

In this style of script generation, we provide objects information directly into the script.

**Example:**

- 1) Invokeapplication "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"
- 2) dialog("text:=Login").Activate
- 3) dialog("text:=Login").Winedit("attached text:=Agent Name:").Set "Naseer"
- 4) dialog("text:=Login").Winedit("attached text:=Password:").Set "mercury"
- 5) dialog("text:=Login").Winbutton("text:=OK", "width:=60").Click

**Note:**

1. Dialog, WinEdit and WinButton – Test Objects
2. text, attached text – Property names
3. Login, Agent Name:, Password:, OK – Property values or Logical Names of the Object
4. Activate, Set, Setsecure, Click – Methods

**Note2:**

If we feel one property information is not sufficient for recognizing the object uniquely, then we can provide more properties information by separating with commas.

**Note 3:**

If we want to get objects information (Test objects, properties and values), we can use object spy feature. This feature is available in Tools Menu, in local repository and in repository manager.

**Example 2:**

```
SystemUtil.Run "C:\Program Files\Internet Explorer\IEXPLORE.EXE", "", "C:\Program Files\Internet Explorer", "open"  
Browser("title:=Google").Page("title:=Google").Sync  
Browser("title:=Google").Navigate "http://www.icicibank.com/"
```

---

If we want maintain 'Objects information' in centralized location then we can use Constants.

Steps:

Creating Constants:

Const Login="text:=Login", Agent="attached text:=Agent Name:"

Const Pwd ="attached text:=Password:", Ok="text:=OK"

Note: we can declare no of Constants in a line by separating with Comas (,), if we take other line then we have to use Const Statement again.

Creating a Library file

Place Constants in Notepad and save as .vbs file

Associate the Library file to QTP (File->Settings->Resources-> Click add (+) icon-> Browse path of the Library file->Click Apply and click Ok buttons

Otherwise, we can load the library file during run-time

Syntax:

ExecuteFile "Path of the Library file(.vbs)"

After that create the Test Script using Constants

Creating the Test Script using Constants:

```
Invokeapplication "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"  
Dialog(Login).Activate  
Dialog(Login).Winedit(Agent).Set "asdf"  
Dialog(Login).Winedit(Pwd").Set "mercury"  
Dialog(Login).Winbutton(Ok).Click
```

Advantages:

If we maintain Object Information in the centralized location, then we can handle modifications easily.

---

## Dynamic Programming

In this style of script generation, first we create description objects, provide properties information and use description objects in the test script.

### Creating Properties Collection Objects

Set oLogin=description.Create

Set oAgent=description.Create

Set oPassword=description.Create

Set oOk=description.Create

### Entering Properties Information into Objects

```
oLogin("text").value="Login"
```

```
oLogin("width").value=320
```

```
oLogin("height").value=204
```

```
oAgent("attached text").value="Agent Name:"
```

```
oPassword("attached text").value="Password:"
```

```
oOk("text").value="OK"
```

### Generating Tests using Properties collection Objects

```
Invokeapplication "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"
```

```
Dialog(oLogin).Activate
```

```
Dialog(oLogin).Winedit(oAgent).Set "Naseer"
```

```
Dialog(oLogin).Winedit(oPassword).Set "mercury"
```

```
Dialog(oLogin).Winbutton(oOK).Click
```

**Note1:** Create Description objects and put into one library file, by associating that library file, we can generate tests.

**Note2:** Dynamic programming is some difficult in preparation than static programming but maintenance is very easy.

---

### Examples:

#### Handling similar/Duplicate Objects using Index Property

```
SystemUtil.Run "C:\Program Files\Internet Explorer\IEXPLORE.EXE"
```



Browser("name:=Google").Page("title:=Google").Sync

Browser("name:=Google").Navigate http://www.jjperfumes.com/

Browser("name:=JJ Perfumes-Discount perfume cheap brand name perfumes, fragrance & cologne online").page("title:=JJ Perfumes-Discount perfume cheap brand name perfumes, fragrance & cologne online").Link("innertext:=Register","index:=0").Click

### **Inserting Checkpoints**

Check point is a verification point, it takes expected result from the user and compares with actual results during execution and provides test results.

**There are 11 Checkpoints available in QTP:**

1. Standard check point
2. Text check point
3. Text area check point
4. Bit map check point
5. Data base check point
6. Accessibility check point
7. XML Check point (from Application)
8. XML Check point (from Resource)
9. Page check point
10. Image checkpoint
11. Table checkpoint

Note 1: From 6 to 11 checkpoints are only for Web.

Note 2: From 9 to 11 checkpoints are hidden checkpoints, we can insert these checkpoints through standard checkpoint.

## **1. Standard Checkpoint:**

It checks object property values. We can use this checkpoint for checking any property value.

### **Navigation for Inserting standard checkpoint**

Keep tool under recording mode > place cursor in desired location > Insert > check point > Standard checkpoint > Show the object > click okay > select property and enter expected results & click Ok and stop Recording.

### **Navigation for Editing standard checkpoint**

Select Checkpoint statement and right click > choose checkpoint properties option > modify the value > click Ok.

### **Navigation for Deleting standard checkpoint:**

Select Checkpoint statements and right click > choose delete option.

### **Inserting Standard check points through active screen:**

View > Active Screen > place cursor in desired location > Place mouse pointer on active screen & right click> choose insert standard checkpoint option > click ok > enter expected result > click ok

Note: Inserting Standard check points through keyword view same as in expert view.

**Note:** We can insert standard checkpoints in 3 ways.

- a. Through expert view,
- b. Through keyword view,
- c. Through Active screen.

## **2) Text Checkpoint:**

It Checks object's text property value in different ways.

### **Navigation:**

Keep tool under Recording mode > Insert menu > checkpoint > Text checkpoint > Show the object > click ok > Select options

(Match case; ignore spaces, exact match, text not displayed.)

We can select one or more options > click ok & stop Recording.

## **3) Text Area Checkpoint:**

It checks the text area present in the application.

### **Navigation:**

Keep tool under Recording mode > Insert menu > Checkpoint > Text area checkpoint > Mark the area of text > select one or more options

(Match case, ignore spaces, exact match, text not displayed.)

Click ok and stop recording.

## **4. Bitmap checkpoint:**

It compares bitmaps; we can compare complete bitmaps as well as part of the bitmaps.

### **Navigation:**

Keep tool under Recording mode > Insert menu > Checkpoint > Bitmap checkpoint > show the Bitmap > click ok > select "check only selected area" option if we want to compare part of the bitmap > click ok > stop recording.

#### **5. Database checkpoint:**

It checks Content of the back end Database.

#### **Navigation:**

Insert > checkpoint > Database checkpoint > choose "specify SQL statement manually" option > click next > click create > select machine data source > Select DSN (QT\_flight32) > click ok > enter SQL statement (select \* from orders) > finish > click ok.

**Note:** here we do not need to put tool under Recording mode and we do not need AUT since data is from backend.

#### **6. Accessibility check point:**

It checks whether the webpage in our web application is developed according to W3C (World Wide Web consortium) Rules and Regulations or not.

It is a configurable checkpoint, according to our requirements, we can customize.

#### **Configuring accessibility checkpoint:**

Tools menu > options > web > advanced > check/uncheck items > click apply > click ok

#### **Invigilation:**

Keep tool under recording mode with web environment > insert > checkpoint > accessibility checkpoint > show the webpage > click ok > click ok > stop recording.

Result Criteria:

- a) If item is available but not according to W3C rules then fail.

- b) If an item is available, according to W3C rules then Pass.
- c) If an item is not available then result would be pass

### **7. XML Check point (from Application)**

It checks content of the XML file.

Navigation:

Keep tool under Recording mode in web environment > insert menu > checkpoint (from application)> show the xml pages >click ok > stop Recording.

### **8. XML Check point (from Resource)**

It checks content of the XML file.

Navigation: Insert menu > checkpoint >xml checkpoint (from resource) > browse path of the XML File > click ok > click ok.

Note: 1. If XML file is an individual and path available, and then we can go for inserting xml checkpoint from resource.

Note: 2. If XML file is part of web application, separate path is not available then we can choose inserting XML checkpoints from application.

### **9. Page checkpoint:**

It checks number of Links, Images and Loading time in a web page.

It is a hidden checkpoint; we can insert this through standard checkpoint.

**Navigation:**

Keep tool under Recording mode with web environment > Insert menu > checkpoint > Standard checkpoint >show the web page > click ok > click ok > stop recording.

#### **10. Image checkpoint:**

It checks Image property values.

#### **Navigation:**

Keep tool under Recording mode with web environment > Insert menu > checkpoint > standard checkpoint > show the image > select image > click ok > click ok > stop recording.

#### **11. Table checkpoint:**

It checks content of the web tables.

Navigation: Keep tool under Recording mode under web environment > Insert menu > checkpoint > standard checkpoint > show the web table > click ok > stop recording.

### **Inserting Output Values**

It is a step in which one or more values are captured during test execution.

The values can later be used as input at a different point in the run session or we can use as input for another test.

It is stored in run time data table; it is located in QTP result window and can be retrieved later.

#### **Types of Output Values:**

1. Standard output value
2. Text output value
3. Text area output value
4. Database output value

5. XML output value (From Application)
6. XML output value (From Resource)

### **1) Standard output value:**

We can use standard output values to output the properties values of most objects.

#### **Navigation:**

Keep tool under Recording mode > Insert menu > output value> standard output value > show the object > click ok > select property > modify the column name (if required) > click ok >click ok >stop recording.

### **2) Text output value:**

We can use Text output values to output text strings displayed in an application. When creating a text output value we can output a part of the objects text and we can also specify text before and after options.

#### **Navigation:**

Keep tool under recording mode> Insert menu> output value > text output value > show the text >select one of the option(output text/text before/text after) > click modify if you want to change the column name > click ok > click ok.> check in runtime table after running it.

### **3) Text Area output value:**

We can use text area output values to output text strings displayed within a defined area of the screen.

#### **Navigation:**

Keep tool under recording mode > insert menu > output value > text area output value >Mark the text area >click modify the column name if necessary > click ok > again click ok > check in run time table after running it..

#### 4) Database output value:

We can use Database output values to output the value of the contents of database cells.

#### Navigation:

Insert menu> output value > Database output value> choose specify SQL statements manually >click next >click create >Select Machine Data source > Select Driver (QT\_flight32) > click ok > enter SQL statement (select \*from orders) > click finish > select data cells > click ok > Run and it will capture and see the results in run time table.

#### 5) XML output value (from application):

we can use XML output values to output the values of XML elements in XML documents.

Navigation: Keep tool under recording mode with web environment > Insert menu > output value > XML output value from application > Show the XML document > select elements > click ok >stop recording.

#### 6) XML output Value (From Resource):

We can use XML output values to output the values of XML elements in XML documents.

#### Navigation:

Insert menu > output value> XML output value from resource > browse path of the XML file > click ok > select elements >click ok.

## Parameterization

Passing parameters, through this we can pass multiple values.



We use parameterization in Data Driven Testing.

[Data Driven Testing](#): Testing the Same operation with multiple sets of test data.

**Types of Parameterization:** We can parameterize tests in several ways in QTP

1. Through Loop Statements
2. Dynamic Test Data Submission
3. Through Data Table
4. Fetching Test Data directly from External files (Flat files & Spreadsheets)
5. Fetching Test Data directly from Databases (MSAccess, Oracle etc).
6. Getting Test Data from front end objects.

**1. Through Loop Statements:** We can use loop statements for passing sequential numbers & Logical Numbers.

Note: We can't generate Strings.

For orderno=1 to 10 step 1 ' for one increment step keyword is not mandatory

Window("Flight Reservation").Activate

Window("Flight Reservation").WinButton("Button").Click

Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set orderno

Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Next

**2.Dynamic Test Data Submission:** Through Loop Statements we can give strings also but every time user has to enter data.

For x=1 to 3

Agent =inputbox("enter an Agent Name")

Password=inputbox("enter a password")

invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set Agent

Dialog("Login").WinEdit("Agent Name:").Type micTab

Dialog("Login").WinEdit("Password:").SetSecure password

Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").Close

Next

**3. Through Data Table:** QTP adds one data table (Spreadsheet) for every test, we can use Data Table for Data Driven Testing.

It has 3 types of usage .

a. Entering test data directly into data table and use

b. Importing test data from external Flat files

c. Importing test data from external Spread sheets

d. Importing test data from Data bases.

**a. Entering test data directly into data table and use.**

**Steps:**

i) Generate the basic test>open data table(View>Data Table)

ii) Click on column header>enter the name of the field (like this we can create number of columns) > Enter Data>connect the data to test

(variable=datatable("column name", Sheet id)

Example: agent=datatable("agent",1)

Pass parameters.)

iii) Run the test.

**Example:**

Agent = Datatable("Agent",1)

pwd=Datatable ("Password",1)

Invokeapplication "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set Agent

Dialog("Login").WinEdit("Agent Name:").Type micTab

Dialog("Login").WinEdit("Password:").SetSecure pwd

Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").Close

**b. Importing test data from external files:**

Open Data Table (view>Data table)>place mouse pointer on data table and right click>file>import from file>Click ok>Browse path of the file(it imports data from the flat file)

Connecting Test Data to QTP Test as above and run the test.

**c. Importing test data from external Spread sheets:**

Open Data Table (view>Data table)>place mouse pointer on data table and right click>file>import from file>Click ok>Browse path of the excel sheet (it imports data from the excel sheet)

Connecting Test Data to QTP Test as above and run the test.

**d. Importing test data from Data bases:**

Through Data table we can import Test Data from Data bases, but first we have to create /get the DSN(Data source Name)& we have to use SQL Commands.

i). Creating a Test Database: open MS Access (or we can use any other database).

ii) Start programs>MS Office>MS Access>file >new>Select blank Database>enter name of the database>Save with mdb extension.

iii)

Creating Tables: Select Create table in design view>Enter field name(Agent)and Select data type(text) Like this we can create number of fields>save&enter table name.

iv)

Entering Data into Tables: Select table>enter the data.

v)

Creating DSN & importing data

**Navigation:**

view>data table>Place mouse pointer on Data table>sheet>import>from database(Database query wizard opens)>choose 'specify SQL statements manually>click next >click create>click new>select driver type>click next >browse path to store> enter DSN Name>Click Save>click next>click finish>select>browse the database& select>click ok>click ok>select DSN>click ok>enter SQL statement (select \*from login)>click finish.

**Note:** DSN Creation is one time activity, by using the DSN we can get data for number of tests.

#### 4. Fetching Test Data directly from Flat files

```
Dim fso, myfile
```

```
Set fso=createobject("scripting.filesystemobject")
```

```
Set myfile=fso.opentextfile("d:\trigun.txt",1)
```

```
myfile.skipline
```

```
While myfile.atendofline <> true
```

```
x=myfile.readline
```

```
S=split(x,"@")
```

```
SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\","open"
```

```
Dialog("Login").Activate
```

```
Dialog("Login").WinEdit("Agent Name:").Set s(0)
```

```
Dialog("Login").WinEdit("Agent Name:").Type micTab
```

```
Dialog("Login").WinEdit("Password:").SetSecure S(1)
```

```
Dialog("Login").WinEdit("Password:").Type micReturn
```

```
Window("Flight Reservation").Close
```

```
Wend
```

### Fetching Test Data directly from Excel Sheets

### Fetching Test Data directly from Databases

Option explicit

Dim con,rs

Set con=createobject("adodb.connection")

Set rs=createobject("adodb.recordset")

con.provider=("microsoft.jet.oledb.4.0")

con.open "C:\Documents and Settings\pooja\My Documents\trigun.mdb"

rs.open "select \* from login",con

do until rs.eof=true

SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\","open"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set rs.fields ("agent")

Dialog("Login").WinEdit("Agent Name:").Type micTab

Dialog("Login").WinEdit("Password:").SetSecure rs.fields("password")

Dialog("Login").WinEdit("Password:").Type micReturn

Window("Flight Reservation").Close

rs.movenext

loop

---

### Using Dictionary Object for Parameterization

Set inputdata=CreateObject("Scripting.Dictionary")

inputdata.Add "Agent","Naseer"

inputdata.Add "Password","mercury"

SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set inputdata("Agent")

Dialog("Login").WinEdit("Password:").Set inputdata("Password")

Dialog("Login").WinButton("OK").Click

# Synchronization

It is a process of matching the speeds of both QTP and AUT in order to get proper execution and results.

## **Where Synchronization is required:**

During test execution QTP gives instructions one by one with same speed, but AUT takes less time for some operations execution and more time for some operations execution, that time we may not get proper execution and results. In order to get proper results in between QTP & AUT synchronization is required.

There are several methods available in QTP for synchronization.

1. Inserting Wait statements.
2. Inserting Synchronization points.
3. Increasing Tool default synchronization time.
4. Sync Method (Only for WEB)
5. Exist Property

## **Inserting Wait Statements:**

We can insert wait statements in our test in order to make QTP to wait until AUT completes current operation.

Syntax: Wait(time in seconds)

**Note:** If we insert wait statements QTP waits up to maximum time even though operation is completed.

## **Inserting Synchronization points:**

Place cursor in desired location>keep tool under recording mode>Insert menu>Synchronization point >show the object >click ok>select property name & Value(True)>enter time in Milli seconds>click ok>Stop recording.

**Note:** if we insert Synchronization points, it does not wait up to maximum time, after completion of the current operations, it goes to next step immediately.

Syntax: object hierarchy.waitproperty "property name",value,time in milli seconds.

#### **Increasing Tool default synchronization time:**

Navigation: File>settings>run tab>increase object synchronization time out>apply>ok

Note: If we increase QTP tool default time that can be applied for all statements in the test, but QTP does not wait up to maximum time unnecessarily, After completion of one statement execution it goes to next statement immediately.

#### **Sync Method:** (only for WEB)

Waits for the browser to complete current navigation.

Syntax: Object Hierarchy.Sync

Ex: Browser("Yahoo! Toolbar").Page("Yahoo! Toolbar").Sync

#### **Selecting an appropriate Method:**

1. Suppose in our test one or more statements only are taking more time for execution then selecting "Inserting synchronization method" is better.
2. Suppose in our test more statements are taking more time for execution then selecting "increasing tool time out" .
3. Above two are local features but wait statement is Vbscript feature, even though some drawbacks are there in using wait statement, it is better to use wait statement in functions.

#### **Advantages of Wait Statement:**

- 1) It is a VBScript statement, we can use this globally
- 2) It is recommended, whenever we want to observe the Test Execution process
- 3) No object reference required to insert wait statements



4) It is useful to make QTP capture values and screen shots during test execution

### Inserting Transaction Points

QTP is providing a Utility object called **Services** for measuring transaction time

Suppose, we want to measure how much time that one test or part of the test is taking for execution/running, we can measure using these transaction point (Start and End)

#### Syntax:

```
Services.StartTransaction "Transaction Name"
```

```
Statements
```

```
_____
```

```
Services.EndTransaction "Transaction Name"
```

Note: these options (Start and End Transactions) available in QTP tool Insert menu

#### Example:

```
services.StartTransaction "Login"
SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\Mercury
Interactive\QuickTest Professional\samples\flight\app\", "open"
Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set "naga"
Dialog("Login").WinEdit("Password:").SetSecure "4baf50f18b0ae0d5f5425fe760653e96da50bde7"
Dialog("Login").WinButton("OK").Click
Window("Flight Reservation").Activate
services.EndTransaction "Login"
Note: We can insert number of transaction points in a test
```

#### Using Timer Function

'Timer' is a VB Script Built-in function, that can be used for measuring transaction time.

ST=Timer' It returns elapsed time since 12:00 AM Mid night in seconds

#### Example:

```
SystemUtil.Run "C:\Program Files\Mercury Interactive\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\Mercury
Interactive\QuickTest Professional\samples\flight\app\", "open"
Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set "naga"
Dialog("Login").WinEdit("Password:").SetSecure "4baf50f18b0ae0d5f5425fe760653e96da50bde7"
Dialog("Login").WinButton("OK").Click
Window("Flight Reservation").Activate
ET=Timer
TT=ET-ST
msgbox TT
```

**Note:** here ST, ET and TT are variables, you can take any variable, I used:

ST for Start Transaction

ET for End transaction

TT for Transaction Time

Debugging Tests

Debugging is a process of executing the script or a program in a User designed fashion with some temporary break points in order to identify the errors.

For debugging, QTP is providing debug commands and breakpoints.

Debug Commands:

**1) Step Into (short cut key F11):**

1. It starts execution
2. It executes one statement at a time.
3. If it is function call, it opens the function, after that it executes one statement at a time.

**2) Step Over (Short cut key F10):**

- a) It executes one statement at a time after execution starts
- b) It executes all functions statements at a time

**Note:** After opening the function, if we use step over it executes one statement at a time only.

**3) Step Out (shift +F11):**

It executes all remaining statements in a function at a time.

Ex: Dim a,b,c

a=100

b=200

c=a+b

msgbox c

msgbox "hello"

Call hyderabad

msgbox "bye bye"

**Note:** Create a script and work on it

#### **Break Point:**

This feature is used for breaking the execution temporarily.

#### **Navigation:**

Place cursor in desired location > Debug menu>Insert/Remove break point

Or

Use short cut key **(F9)**

Or

Place mouse pointer before the statement and click.

Note: If we want to execute Some part of the Test Script at a time and some other part of the test script step by step, there we can use breakpoints.

#### **Debug viewer Pane:**

It is used for viewing, modifying or setting the value of the variables during the execution break with the help of 3 Tabs.

1. Watch
2. Variables
3. Command

Note: Through Debug Viewer Pane, We can watch variables whether the variables are taking correct values or not at a specific point of time. We can change Values of Variables temporarily.

Ex :

Dim x,y,z

x=10

y=20

z=x+y

msgbox z

msgbox "Hello"

x=35

y=67

z=x\*y

msgbox z

msgbox "bye bye"

Navigation: View menu > Debug Viewer > Use step into feature.

## Analyzing Test Results & Reporting

Analyzing the Test Result is not a responsibility of QTP. After Running Tests, User (Test Engineer) has to analyze the test result. and send defect reports.

### **QTP Result Window:**

After Running/Executing a test, then QTP Provide result window. we export test result from result window in Document (.Doc), HTML, XML and PDF formats.

Note: QTP doesn't provide batch wise test result, if we execute test batches. It provides test wise result only.

### **Status of the Result:**

QTP has 4 types of result status

- a) Pass
- b) Fail
- c) Done
- d) Warning

### **Defining our own Result:**

Generally QTP provides, test results, if we use qtp tool features like checkpoints. suppose if we use flow control statements (conditional and Loop statements), then we have to define our own result.

For defining our own results, QTP is providing an utility object called "**Reporter**"

by using reporter, we can define our own results.

Syntax:

Reporter.ReportEvent status of the Result,"Result Step","message"

Example"

Reporter.ReportEvent micPass,"Res","Login operation Successful"

explanation:

Reporter- Utility object

ReportEvent-Method

micPass-Status of the Result.

Res-Result step

"Login operation Successful"- Message

### **Status of the Result:**

- 1) Pass -micPass or 0
- 2) Fail -micPass or 1
- 3) Done- micDone or 2
- 4) Warning- micDone or 3

### **Environment Variables in QTP**

These are global variables; if you want to use some variables globally (From number of tests) declaring as global variables is better. If we declare as local variables in number of tests, modifications are difficult.

### **Types of Environment variables:**

1. **Built in variables:** These are system defined variables, we can access from all tests but these are designed as read only.

2. **User defined variables:** Variables that we predefine in the active external environment variables file. We can create as many files as we want and we can change values of variables.

### **Usage of environment variables:**

Syntax:

Variable = Environment("environment variable name")

Ex:

X=Environment ("ActionName")

Msgbox x

### **Example Script:**

ProductDir =environment ("ProductDir")

app= "\samples\flight\app\flight4a.exe"

Systemutil.Run ProductDir & app

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set "kajal"

Dialog("Login").WinEdit("Password:").SetSecure "4b3c86f2107ff565cc195ba6c24739091b656407"

Dialog("Login").WinButton("OK").Click

### **Creating user defined environment variables:**

Navigation:

File > settings > Environment > select variable type as user defined > click add icon (+) > enter variable name & Value > click ok (like this we can create number of variables) > click export > browse path & enter file name, save with xml extension > click ok.

### **Associating environment variable file:**

Navigation:

File > settings > Environment > select variable type as user defined > check "load variables and values from the external file" > browse path of the xml file > click apply & ok.

Or

### **We can load environment file directly**

Environment.loadfromfile"path of the xml file"

### **Usage of user defined Environment variables:**

Associate environment file to current test:

Variable = Environment ("Variable Name")

X=environment ("city")

Msgbox x

### **Modifying Environment files:**

Select XML file>open with notepad>modify the values>Save.

## **Regular Expressions in QTP**

### **What is Regular Expression?**

It is a way of representing data using symbols. They are often used within matching, searching or replacing algorithms.

### **Regular Expressions in QTP:**

Regular expressions can be used in QTP for identifying objects and text strings with varying values.

### **Where we use:**

- o Defining the property values of an object in Descriptive programming for handling dynamic objects
- o For parameterizing a step
- o creating checkpoints with varying values

### **Using Regular Expressions in QTP:**

We can define a regular expression for a constant value, a Data Table parameter value, an Environment parameter value, or a property value in Descriptive programming.

We can define a regular expression in standard checkpoint to verify the property values of an object; we can set the expected value of an object's property as a regular expression so that an object with a varying value can be verified.

We can define the text string as a regular expression, when creating a text checkpoint to check that a varying text string is displayed on our application, For XML checkpoints we can set attribute or element values as regular expressions.

### **Ways of Regular Expressions:**

#### **a) Backslash Character:**

A backslash (\) can serve two purposes. It can be used in conjunction with a special character to indicate that the next character be treated as a literal character.

Alternatively, if the backslash (\) is used in conjunction with some characters that would otherwise be treated as literal characters, such as the letters n, t, w, or d, the combination indicates a special character.

#### **b) Matching Any Single Character:**

A period (.) instructs QTP to search for any single character (except for \n).

Ex:



welcome.

Matches welcomes, welcomed, or welcome followed by a space or any other single character.

**c) Matching Any Single Character in a List:**

Square brackets instruct QTP to search for any single character within a list of characters.

Ex:

To search for the date 1867, 1868, or 1869, enter:

186[789]

**d) Matching Any Single Character Not in a List:**

When a caret (^) is the first character inside square brackets, it instructs QTP to match any character in the list except for the ones specified in the string.

Example:

[^ab]

Matches any character except a or b.

**e) Matching Any Single Character within a Range:**

To match a single character within a range, we can use square brackets ([ ]) with the hyphen (-) character.

Example:

For matching any year in the 2010s, enter:

201[0-9]

**f) Matching Zero or More Specific Characters:**

An asterisk (\*) instructs QTP to match zero or more occurrences of the preceding character.

For example:

ca\*r

Matches car, caaaaaar, and cr

**g) Matching One or More Specific Characters:**

A plus sign (+) instructs QTP to match one or more occurrences of the preceding character.

For example:

ca+r

Matches car and caaaaaar, but not cr.

**h) Matching Zero or One Specific Character:**

A question mark (?) instructs QTP to match zero or one occurrences of the preceding character.

For example:

ca?r

Matches car and cr, but nothing else.

**i) Grouping Regular Expressions:**

Parentheses (()) instruct QTP to treat the contained sequence as a unit, just as in mathematics and programming languages. Using groups is especially useful for delimiting the argument(s) to an alternation operator (|) or a repetition operator (\*, +, ?, {}).

**j) Matching One of Several Regular Expressions:**

A vertical line (|) instructs QTP to match one of a choice of expressions.

**k) Matching the Beginning of a Line:**

A caret (^) instructs QTP to match the expression only at the start of a line, or after a newline character.

l) Matching the End of a Line:

A dollar sign (\$) instructs QTP to match the expression only at the end of a line, or before a newline character.

**m) Matching Any AlphaNumeric Character Including the Underscore:**

\w instructs QTP to match any alphanumeric character and the underscore (A-Z, a-z, 0-9, \_).

**n) Matching Any Non-AlphaNumeric Character:**

\W instructs QTP to match any character other than alphanumeric characters and underscores.

**o) Combining Regular Expression Operators:**

We can combine regular expression operators in a single expression to achieve the exact search criteria we need.

For example,

start.\*

Matches start, started, starting, starter, and so forth.

we can use a combination of brackets and an asterisk to limit the search to a combination of non-numeric characters.

For example:

[a-zA-Z]\*

To match any number between 0 and 1200, we need to match numbers with 1 digit, 2 digits, 3 digits, or 4 digits between 1000-1200.

The regular expression below matches any number between 0 and 1200.

([0-9]?[0-9]?[0-9]?1[01][0-9][0-9]1200)

**RegExp object**

VB Script is providing RegExp object for defining Regular expressions, It provides simple support for defining regular expressions.

Regular Expression Object Properties and Methods:

**Properties:**

a) Global Property

b) IgnoreCase Property

c) Pattern Property

**Methods:**

a) Execute Method

b) Replace Method

c) Test Method

## Regular Expressions Examples:

### 1) Match File Names in a Directory against Regular Expression

```
Set objFS = CreateObject("Scripting.FileSystemObject")
Set objShell = CreateObject("WScript.Shell")
strCurrentDirectory = objShell.CurrentDirectory
```

```
Set objFolder = objFS.GetFolder(strCurrentDirectory)
Set colFiles = objFolder.Files
```

```
Set objRE = New RegExp
objRE.Global = True
objRE.IgnoreCase = False
objRE.Pattern = WScript.Arguments(0)
```

```
For Each objFile In colFiles
bMatch = objRE.Test(objFile.Name)
If bMatch Then
WScript.Echo objFile.Name
End If
Next
```

### 2) Match Content in a File against a Regular Expression

```
strFileName = "E:\Naseer.txt"
Set objFS = CreateObject("Scripting.FileSystemObject")
```

```
Set objTS = objFS.OpenTextFile(strFileName)
strFileContents = objTS.ReadAll
WScript.Echo "Searching Within: "
WScript.Echo strFileContents
objTS.Close
```

```
Set objRE = New RegExp
objRE.Global = True
objRE.IgnoreCase = False
objRE.Pattern = WScript.Arguments(0)
```

```
Set colMatches = objRE.Execute(strFileContents)
WScript.Echo vbNewLine & "Resulting Matches:"
For Each objMatch In colMatches
WScript.Echo "At position " & objMatch.FirstIndex & " matched " & objMatch.Value
Next
```

## Batch Testing or Batch Execution

Executing a group of Tests or series of tests at a time is known as Batch Testing or Batch Execution. For performing Batch Testing, QTP is providing a separate Tool called Test Batch Runner.

### Steps for Batch Testing:

- 1). Create Individual Tests and Run once.
- 2). Open 'Test batch Runner' Tool and Form Batches.
- 3) Provide permission to 'Test batch Runner' to run tests
- 4). Run or Execute Test Batches from Test Batch Runner.

5) View Test wise Result in 'Test Result Viewer'.

**Note:** QTP doesn't provide Batch wise result.

#### **Forming a Test Batch:**

##### **Launch Test Batch Runner.**

Navigation: Start>program>quicktest professional>Tools>Test Batch Runner>File>new>batch>add>browse path of the test (like this add number of tests)>save with MTB extension (Module test batche)>close test batch runner.

#### **Running or Executing a Test Batch**

Open Test Batch Runner Tool

Navigation: File>open>browse path of the test batch>batch>run

Note: Test Batch Runner launches QTP Tool, QTP runs Tests one by one.

Note: Allowing other products to Run Tests.

Navigation: Tools>Options>Run>check allow other mercury products>apply & Ok

#### **Executing partial Test Batch**

Open Test Batch Runner >open Test Batch>Select or deselect tests>run test batch.

#### **Recovery Scenario Manager**

To recover from unexpected events and errors that are occurred in the test environment during run session, we can use Recovery Scenario Manager. For good recovery, error must be known the occurrence is unknown.

#### **There are (4) Types of events such as:**

##### **(i) Application Crash**

An open application fails during Test Run.

#### **Navigation:**

Resources Menu -> Recovery Scenario Manager- > Click New- > Click Next – >

Select Application Crash as Trigger event->Next – >Select selected executable

application->Next ->Select Recovery Operation [Keyboard, Mouse Operation,  
Close Application Process, function Call, Restart, Microsoft Windows] ->Next –>If  
you want to check Add another operation else uncheck->Next –>Next –>Enter  
Scenario Name ->Next- >Select Option –>Finish –>Close –>Save the scenario in  
specified location with “.qrs”  
(qrs stands for QuickTest Recovery Scenario.)

## **(ii) Popup Window.**

To handle unwanted popups.

### **Navigation:**

Resources Menu ->Recovery Scenario Manager ->New –>Next –>Select “Popup  
Window” as Trigger event ->Next –>Click on Hand Icon –>Show unwanted  
window with Hand icon ->Next –>Next –>Select function call as Recovery  
Operation ->Next [Open Notepad ->Save empty file with .vbs extension] –>Browse  
the .vbs file path ->Next –>Uncheck Add another Recovery Operation –>Next –>  
Select Post-Recovery Test Run Option [Repeat current step and continue, Proceed to  
Next step, Proceed to Next Action, Proceed to next test iteration, Restart current test  
run, Stop the Test Run] ->Next –>Enter Scenario Name ->Next –>Select Option –>  
Finish ->Save the scenario with “.qrs” ->Record required Recovery Operation [Click  
ok, Click Cancel] take the script into function ->Save the library file ->Click Run

## **(iii) Test Run Error.**

A step in your test does not run successfully then Test Run Error  
can be raised.

#### Navigation :

Resources Menu -> Recovery Scenario Manager – > New – > Next – > Select "Testrunerror

Window" as Trigger event -> Next -> select any error o -> Next – > Next – > Select function call as Recovery

Operation -> Next [Open Notepad - > Save empty file with .vbs extension] -> Browse

the .vbs file path -> Next – > Uncheck Add another Recovery Operation – > Next ->

Select Post-Recovery Test Run Option [Repeat current step and continue, Proceed to

Next step, Proceed to Next Action, Proceed to next test iteration, Restart current test

Select Option ♦♦ Next ♦♦ Enter Scenario Name ♦♦ Next ♦♦ run, Stop the Test Run] ♦♦

Record required Recovery Operation [Click Save the scenario with ".qrs" ♦♦ Finish ♦♦

Click Run Save the library file ♦♦ ok, Click Cancel] take the script into function ♦♦

#### (iv) Object State.

The property values of an object in your application match

specified values. You can specify property values for each object in the

hierarchy.

#### Navigation:

Resources Menu -> Recovery Scenario Manager -> New -> Next -> Select "Object state

Window" as Trigger event -> Next -> Click on Hand Icon -> Show object with hand icon

-> Next -> Next->select object property with value (enabled ,false)->click next -> Select function call as Recovery

Operation -> Next [Open Notepad -> Save empty file with .vbs extension] -> Browse

the .vbs file path -> Next -> Uncheck Add another Recovery Operation -> Next ->

Select Post-Recovery Test Run Option [Repeat current step and continue, Proceed to

Next step, Proceed to Next Action, Proceed to next test iteration, Restart current test

run, Stop the Test Run] -> Next-> Enter Scenario Name -> Next -> Select Option ->

Finish -> Save the scenario with ".qrs" -> Record required Recovery Operation [Click

ok, Click Cancel] take the script into function -> Save the library file -> Click Run

# Automation Object Model

## **AOM Scripting (Automation Object Model)**

### **Object Model:**

An object model is a structural representation of software objects (classes) that comprise the implementation of a system or application. An object model defines a set of classes and interfaces, together with their properties, methods and events, and their relationships.

We can use QTP Automation object Model to write scripts, that automate our QTP operations.

QTP Automation object model provides objects, methods and properties that enable us to control QTP from another application.

We can use Scripting languages or programming languages such as VBscript, Java script or VC++, .Net for automating QTP operations.

Example:

### **1) Write an AOM Script to launch QTP Tool, Execute Tests and to close the QTP Tool**

option explicit

Dim qtApp

Set qtApp=createobject ("Quicktest.Application")

qtApp.Launch

qtApp.visible=True

qtApp.open "C:\Documents and Settings\admin\My Documents\login"

```
qtApp.Test.Run
```

```
qtApp.Test.Close
```

```
qtApp.open "C:\Documents and Settings\admin\My Documents\open order"
```

```
qtApp.Test.Run
```

```
qtApp.Test.Close
```

```
qtApp.quit  
Set qtApp=Nothing
```

Note: AOM Script can be used in Automation framework as an Initialization script.

---

## 2) Write an AOM script to execute series of tests

```
Dim objFso,objQtp,myFile,i,Naseer
```

```
Set objQtp = CreateObject("Quicktest.Application")  
objQtp.Visible = True  
objQtp.Launch
```

```
Set objFso = CreateObject("Scripting.FileSystemObject")  
Set myFile = objFso.OpenTextFile("C:\Users\Kiran\Desktop\Test\Naseer.txt")  
i = 1  
Do Until myFile.AtEndOfStream = True  
testurl = myFile.ReadLine
```

```
objQtp.Open Naseer  
objQtp.Test.Run  
objQtp.Test.Close  
i = i+1  
Loop
```

```
objQtp.Quit  
Set objQtp = Nothing
```

## Dynamic handling of object Repositories

Loading repositories during running, finding path of the repositories and removing repositories is called Dynamic Handling of Object Repositories.

Using this feature we can increase QTP performance. To do this, QTP is providing an object called "**RepositoriesCollection**".



**Syntax for Loading a Repository:**

```
RepositoriesCollection.Add "Path of the Repository File"
```

**Syntax for finding Position of the Repository:**

```
Variable=RepositoriesCollection.Find("Path of the Repository")
```

**Syntax for Removing the Repository:**

```
RepositoriesCollection.Remove(position)
```

**Syntax for Removing All Repositories:**

```
RepositoriesCollection.RemoveAll
```

**Example:**

```
RepPath="C:\Documents and Settings\Administrator\My Documents>Login.tsr"
```

```
RepositoriesCollection.Add (RepPath)
```

```
systemutil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"
```

```
Dialog("Login").Activate
```

```
Dialog("Login").WinEdit("Agent Name:").Set "sudhakar"
```

```
Dialog("Login").WinEdit("Password:").Set "mercury"
```

```
Dialog("Login").WinButton("OK").Click
```

```
pos=RepositoriesCollection.Find(RepPath)
```

```
RepositoriesCollection.Remove(pos)
```

```
RepositoriesCollection.RemoveAll
```

**Automation Framework****Automation:**

Accessing one Software Objects from another Software is called Automation.

(AUT is one s/w and Testing Tool is another s/w.)

#### **Automation Framework:**

It is a set of guidelines, assumptions and process developed by experts in order to perform a task(s) in an effective, efficient and optimized way.

A systematic approach for automating software applications.

#### **Why Automation Framework:**

In one project test automation we use various files, we perform various tasks, in order to organize and manage them all, a systematic approach (Automation Framework) required.

#### **files to be used:**

a) Flat files (.txt),

b) Excel files (.xls)

c) Object repository files (.tsr)

d) XML files (.xml)

e) Library files (.vbs)

f) Test Scripts (.mts)

g) Test Batch Files (.mtb)

h) Recovery scenario files (.qrs)

i) Test Result file (.XML) Etc...

### **Tasks to be performed:**

#### **a) Analyzing the application**

We analyze the AUT in terms of Object Identification and Functionality. Basically, QTP is an Object based test tool, here first we have to confirm if the QTP is recognizing all the objects in our application or not. To do this, we can follow one or more below given methods...

- i) Record and playback some End to End Scenarios.
- ii) Using Object Spy
- iii) Using Object Repository Manager

We can segregate Software Objects in 3 categories...

- i) Static Objects(Properties information is constant)
- ii) Dynamic Objects(Properties info changes dynamically)
- iii) Duplicate Objects(Having Similar property info.)

Note:We can handle dynamic Objects using Regular expressions and handle duplicate Objects using "Index" property.

#### **b) Selecting Areas/Test Cases for Automation**

Overview:

100% test automation is not possible, we can go for maximum automation.

- i)Tests that we want to execute on multiple versions of software(Sanity,Regression Tests)
- ii)Tests that we want to execute with multiple sets of test data.(Data Driven Tests)
  - iii) Tests having complex functionality

c) Planning (Effort estimation,Schedule,Work allocation and Automation Framework implementation)

Overview:

After selecting some areas for Automation, we have to estimate the efforts in terms of time and resources(Environmental and Human)

To estimate efforts, we can follow an one of the below given methods:

- a) Using Expert based techniques.
- b)Experienced based techniques.
- c)Function point analysis.

- 1) Identifying End to End Scenarios.
- 2) Identifying verification points.
- 3) Identifying General and Alternate Flows.
- 4) Identifying the modularity.
- 5) Identifying the complex functionality.

d) Creating Generic and project specific functions.

e) Creating required Automation resources(Functions,Object repository Files,Environment variable files—etc)

f) Creating Tests(Using Object Repository or Descriptive Programming)

g) Enhancing Tests(Inserting Checkpoints,Output Values,Transaction points,Flow Control Statements,Synchronization,Parametrization—etc )

h) Debugging and running tests(Individual test execution,Test Batch execution –etc)

i) Defining and exporting test results

j) Analyzing results

k) Test reporting

l) Defect Reporting using companies defect management system.

m) Modifying tests

n) Performing re and regression testing —etc

**Types Of Automation Framework:**

Automation Framework is not a qtp feature, it's a 3rd party concept. And this is purely local concept.(framework may vary from one company to another)

1. **Record/Playback** or Linear Framework (1st generation framework).

In this Framework we create tests using recording,low resources maintenance, Form batches and executing tests.

Drawbacks: Modifications and maintenance are difficult,less command on scripts,low performance —etc

2. **Modular Framework.**

In this Framework,reusable components can be recognized,first we concentrate on creating reusable components, then creating tests using those components.

**Advantages:**

It reduces test Automation time

Performance is high than recording and playback

**Drawbacks:**

Less usage of keywords,no concentration on Data driven testing and low performance than latest Frameworks like Keyword Driven Framework.

3. **Data Driven Framework**

In this Framework,we concentrate more on Data Driven testing.We recognize positive and negative scenarios,then collecting test data and parameterizing.

**Advantages:**

We can check reliability of the system,positive and negative testing.

**Drawbacks:**

Less concentration on complex Functionality Testing and low keywords usage.

4. **Keyword Driven framework**

5. **Hybrid Framework**

In the above frameworks Keyword Driven framework is very famous in the industry.

It is a mixing of more than one approach.

In this approach,we mix Data driven and Modular approaches OR Data Driven and Keyword Driven approaches—etc

**Advantages:**

Scope is very high as we mix different approaches,Flexible for performing any tasks.

**Drawbacks:**

Organizing and managing resources are difficult, complex architecture, low in performance.

**NOTE:** Now a days Mix of Data driven and Keyword driven approaches is a famous Hybrid Framework in the industry.

**Keyword Driven Framework****Keyword:**

1. Any word used as the key to a code

2. A reserved word used to identify a specific command, function etc.

(in our test automation example keywords are :

**Keywords to be used in Test Automation:**

a) Test Objects (EX: Browser, page, webEdit, WinButton—etc)

b) Functions (Built-In and User defined)

c) Methods (EX: Set, Select, Activate, Navigate—etc)

d) Statements (EX: Dim, If, For—etc)

e) Operators (EX: +, /, \*, and, or, not—etc)

**Keyword Driven Approach:**

In this approach we use keywords for preparing tests. First we create tests and make them as functions, through framework we execute them and generate results.

**Key elements of Automation Framework:**

1. Well defined folder structure
2. Initialization script
3. Driver script
4. input data spreadsheet
5. process guidelines document

**Why folder structure?**

In order to create, store, organize and manage files a well defined folder structure required. Folder structure is a mandatory element of any framework, but folder names may vary from one framework to another and company to another .

**Folder Structure:**

1. Object Repository
2. Environment
3. Library
  - a) Company
  - b) Project Specific
4. Test data

5. Test log

6. Recovery scenarios

7. Miscellaneous

And

1. Initialization script (.vbs file)

2. Driver script (QTP Script file)

1) **Object Repository:** In this folder we store object repository files(.tsr), all team members can use this repositories.

2) **Environment:** In this folder we store environment variables files(.xml), all team members can use this variables.

3) **Function Library:** this folder contains two sub folders one is for storing common functions of our company, another folder for storing our project specific functions(.vbs).

4) **Test Data:** in this folder we store test data required for data driven testing, for that we use either .xls files or .txt files or .mdb files.

5) **Test Log:** In this folder we store test result(excel sheet).

6) **Recovery scenarios:** In this folder we store qtp recovery scenarios files(.qrs).

7) **Miscellaneous:** in this folder we store the files other than above files(ex-process guideline doc and messages among team members and instructions by team lead etc).

And



1) **Initialization script** (.vbs file): It launches qtp tool and calls driver script, at end it closes qtp tool.

2) **Driver script** (QTP Script file): It is only the qtp script, we can associate all resources to this script

## Hybrid Framework

**Hybrid Framework:** It is a mixture of two or more approaches

---

To explain this Hybrid Framework, I have taken QTP sample Application (**Flight Reservations**)

---

### Process Guidelines:

---

i) Creating the Folder structure

ii) Creating Automation Resources

iii) Create Organizer spread sheet

Organizer provides instructions to the Driver

iv) Generating the driver Script (QTP Test)

(We associate all resources to the Driver

Driver executes tests based on Organizer spread sheet instructions)

v) Generate the initialization Script (AOM Script file)

It launches the QTP Tool, Calls the Driver Script

Driver executes tests one by one

It closes the QTP Tool

---

#### Test Scenarios:

**Click on this Excel sheet Image in order to get Big size and Clear Image**

#### Hybrid Framework Functions:

```
*****
'
    Login Operation
*****
Function Login(Agent, Password)
SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"
Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set Agent
Dialog("Login").WinEdit("Password:").Set Password
Dialog("Login").WinButton("OK").Click
If Window("Flight Reservation").Exist(12) Then
'Window("Flight Reservation").Close
Login="Login Operation Successful"
'Msgbox Login
Else
Systemutil.CloseDescendentProcesses
Login="Login Failed"
'Msgbox Login
End if
End Function
*****
'
    Open Order
*****
Function Open_Order(ord)
Window("Flight Reservation").Activate
Window("Flight Reservation").WinButton("OpenOrder").Click
Window("Flight Reservation").Dialog("Open Order").Activate
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("OrderNum").Set ord
Window("Flight Reservation").Dialog("Open Order").WinButton("text:=OK").Click
OrderNum=Window("Flight Reservation").WinEdit("Order No:").GetROProperty("text")
OrderNum=Cint(OrderNum)
If ord=OrderNum Then
Open_Order=ord&" Order Opened Successfully"
'Msgbox Order_Number
Else
Open_Order=ord&" Order Not Opened"
'Msgbox Order_Number
End If
```

```

End Function
'*****
'
'      Update Order
'*****

Function Update_Order(Tickets)
Window("Flight Reservation").Activate
Window("Flight Reservation").WinButton("OpenOrder").Click
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("OrderNum").Set "5"
Window("Flight Reservation").Dialog("Open Order").WinButton("text:=OK").Click
Window("Flight Reservation").WinEdit("Tickets:").Set Tickets
Window("Flight Reservation").WinButton("Update Order").Click
Wait 10
Message=Window("Flight Reservation").ActiveX("Threed Panel Control").GetROProperty("text")
If Message="Update Done..." Then
Update_Order="Order Updated Sucessfully"
'Msgbox Update_Order
Else
Update_Order="Order Not Updated"
'Msgbox Update_Order
End If
End Function
'*****
'
'      Close Application
'*****

Function Close_App()
If Window("Flight Reservation").Exist(3) Then
Window("Flight Reservation").Close
End If
End Function
'*****
'
'      Login for Data Driven Testing
'*****

Function Login2(Agent, Password)
SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"
Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set Agent
Dialog("Login").WinEdit("Password:").Set Password
Dialog("Login").WinButton("OK").Click
If Window("Flight Reservation").Exist(12) Then
Window("Flight Reservation").Close
Login2="Login Operation Sucessful"
'Msgbox Login
Else
Systemutil.CloseDescendentProcesses
Login2="Login Failed"
'Msgbox Login
End if
End Function
'*****
'
'      Open Order for Data Driven Testing
'*****

Function Open_Order2(ord)
Window("Flight Reservation").Activate
Window("Flight Reservation").WinButton("OpenOrder").Click
Window("Flight Reservation").Dialog("Open Order").Activate
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("OrderNum").Set ord
Window("Flight Reservation").Dialog("Open Order").WinButton("text:=OK").Click
OrderNum=Window("Flight Reservation").WinEdit("Order No:").GetROProperty("text")
OrderNum=Cint(OrderNum)
If ord=OrderNum Then
Open_Order2=ord&" Order Opened Sucessfully"
'Msgbox Order_Number
Else
Open_Order2=ord&" Order Not Opened"
'Msgbox Order_Number
End If

```

```

End Function
'*****
'      Update Order for Data Driven Testing
'*****
Function Update_Order2(Tickets)
Window("Flight Reservation").Activate
Window("Flight Reservation").WinButton("OpenOrder").Click
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("OrderNum").Set "5"
Window("Flight Reservation").Dialog("Open Order").WinButton("text:=OK").Click
Window("Flight Reservation").WinEdit("Tickets:").Set Tickets
Window("Flight Reservation").WinButton("Update Order").Click
Wait 10
Message=Window("Flight Reservation").ActiveX("Threed Panel Control").GetROProperty("text")
If Message="Update Done..." Then
Update_Order2="Order Updated Successfully"
'Msgbox Update_Order
Else
Update_Order2="Order Not Updated"
'Msgbox Update_Order
End If
End Function

```

### Organizer Spread Sheet: I st Sheet "Module"

**Click on this Excel sheet Image in order to get Big size and Clear Image**

### II nd Sheet "TestCase"

**Click on this Excel sheet Image in order to get Big size and Clear Image**

### III rd Sheet "TestStep"

**Click on this Excel sheet Image in order to get Big size and Clear Image**

### Test Data:

**Click on this Excel sheet Image in order to get Big size and Clear Image**

### Driver Script:

'Adding sheets to Run-time data table

DataTable.AddSheet "Module"

DataTable.AddSheet "TestCase"

DataTable.AddSheet "TestStep"

DataTable.AddSheet "Login"

'Importing data from External File

DataTable.ImportSheet "C:\Documents and Settings\gcr\Desktop\Proj\_Automation\Organizer\Organizer.xls",1,3

DataTable.ImportSheet "C:\Documents and Settings\gcr\Desktop\Proj\_Automation\Organizer\Organizer.xls",2,4

DataTable.ImportSheet "C:\Documents and Settings\gcr\Desktop\Proj\_Automation\Organizer\Organizer.xls",3,5

DataTable.ImportSheet "C:\Documents and Settings\gcr\Desktop\Proj\_Automation\TestData\data.xls",1,6

'Capturing Executable Modules from Module Sheet

MRowCount=DataTable.GetSheet("Module").GetRowCount

For i=1 to MRowCount Step 1

DataTable.SetCurrentRow(i)

ModuleExe=DataTable(3,"Module")

If UCase(ModuleExe)="Y" Then

ModuleId=DataTable(1,"Module")

'Msgbox "ModuleId: " & ModuleId

'       Capturing executable test cases under executable modules

TCRowCount=DataTable.GetSheet("TestCase").GetRowCount

For j=1 to TCRowCount Step 1

DataTable.SetCurrentRow(j)

TCCaseExe=DataTable(3,"TestCase")

ModuleId2=DataTable(4,"TestCase")

If UCase(TCCaseExe)="Y" and ModuleId=ModuleId2 Then

TestCaseId=DataTable(1,"TestCase")

'Msgbox TestCaseId

'Capturing keywords for executable steps

TSRowCount=DataTable.GetSheet("TestStep").GetRowCount

For k=1 to TSRowCount Step 1

DataTable.SetCurrentRow(k)

TestCaseId2 = DataTable(5,"TestStep")

If TestCaseId=TestCaseId2 Then

keyword=DataTable(4,"TestStep")

'msgbox keyword

Select Case keyword

Case "In"

Result=Login("abcd","mercury")

DataTable(7,"TestStep")=Result

Case "oo"

Result=Open\_Order(5)

DataTable(7,"TestStep")=Result

Case "uo"

Result=Update\_Order(5)

DataTable(7,"TestStep")=Result

Case "ca"

Close\_App()

Case "Ind"

Rows=DataTable.GetSheet("Login").GetRowCount

For m=1 to Rows Step 1

DataTable.SetCurrentRow(m)

Result=Login2(DataTable(1,"Login"),DataTable(2,"Login"))

DataTable(3,"Login")=Result

Next

Case "ood"

Rows=DataTable.GetSheet("Login").GetRowCount

For n=1 to Rows Step 1

DataTable.SetCurrentRow(n)

Result=Open\_Order2(DataTable(4,"Login"))

DataTable(5,"Login")=Result

Next

Case "uod"

Rows=DataTable.GetSheet("Login").GetRowCount

For p=1 to Rows Step 1

DataTable.SetCurrentRow(p)

Result=Update\_Order2(DataTable(6,"Login"))

DataTable(7,"Login")=Result

Next

End Select

End If

Next

End If

Next

End If

Next

```
DataTable.ExportSheet "C:\Documents and Settings\gcr\Desktop\Proj_Automation\TestResult\Result1.xls","TestStep"
```

```
DataTable.ExportSheet "C:\Documents and Settings\gcr\Desktop\Proj_Automation\TestResult\Result1.xls","Login"
```

#### **Initialization Script:**

```
Dim objQTP
```

```
Set objQTP=CreateObject("QuickTest.Application")
```

```
objQTP.Visible=True
```

```
objQTP.Launch
```

```
objQTP.Open "C:\Documents and Settings\gcr\Desktop\Proj_Automation\Drivers\Driver1"
```

```
objQTP.Test.Run
```

```
objQTP.Test.Close
```

```
objQTP.Quit
```

```
Set objQTP=Nothing
```

## Web Scripts

### **1) Count all opened Browsers on desktop and close them all?**

```
Set oDesc = Description.Create()
```

```
oDesc("micclass").Value = "Browser"
```

```
Set Browsers =Desktop.ChildObjects (oDesc)
```

```
NumberOfBrowsers = Browsers.Count()
```

```
Reporter.ReportEvent 2,"Res","Number of Browsers are: "&NumberOfBrowsers
```

```
For Counter=0 to NumberOfBrowsers-1
```



```
Browsers(Counter).Close
```

```
Next
```

## **2) Count, how many links available in Mercury Tours Home Page.**

```
Set oDesc = Description.Create()
```

```
oDesc("micclass").Value = "Link"
```

```
Set Lists = Browser("Welcome: Mercury").Page("Welcome: Mercury").ChildObjects(oDesc)
```

```
NumberOfLinks = Lists.Count()
```

```
Reporter.ReportEvent 2,"Res","Number of Links are: "&NumberOfLinks
```

## **3) Verify whether the 'Gmail' link available or not on Google Homepage**

```
Option explicit
```

```
Dim oLink, Links, TotLinks, i, myLink
```

```
Set oLink = description.Create
```

```
oLink("micclass").value = "Link"
```

```
SystemUtil.Run "D:\Program Files\Internet Explorer\IEXPLORE.EXE"
```

```
set Links = Browser("name:=Google").page("title:=Google").ChildObjects(oLink)
```

```
TotLinks = Links.count
```

```
For i =0 to TotLinks-1
```

```
myLink = Links(i).getproperty("innertext")
```

```
If mylink = "Gmail" Then
```

```
reporter.ReportEvent 0,"res","Link Gmail available"
```

```
End If
```

```
Next
```

## **4) Count number of Links, Edit Boxes available on Google Homepage?**

```
Dim oLink
```

```
Set oLink=description.Create
```

```
Set oEdit=description.Create
```

```
oLink("micclass").value="Link"
```

```
oEdit("micclass").value="WebEdit"
```

```
Set Links=browser("name:=Google").page("title:=Google").ChildObjects(oLink)
```

```
Set EditBoxes=browser("name:=Google").page("title:=Google").ChildObjects(oEdit)
```

```
TotLinks= Links.count
```

```
TotEditBoxes=EditBoxes.count
```

```
msgbox TotLinks
```

```
msgbox TotEditBoxes
```

```
Reporter.ReportEvent 0,"Result","Total Links are: "&TotLinks&"Total edit Boxes are: "&TotEditBoxes
```

## **5) Count how many links available in a Web Page(any web page, using Regular Expressions)**

```
Dim oLink,Links, TotLinks
```

```
Set oLink=Description.Create
```

```
oLink("micclass").value="Link"
```

```
Set Links=Browser("title:=.*").page("title:=.*").ChildObjects(oLink)
```

```
TotLinks=Links.count
```

```
msgbox TotLinks
Reporter.ReportEvent 2,"Res","Total Links are: "&TotLinks
```

#### **6) Verify Cost in jjperfumes.com**

##### **'Test Flow**

##### **'i) Launch jjperfumes.com**

##### **'ii) Select a product and enter some quantity**

##### **'iii) select Add to Cart, capture Unit price, Quantity and Cost**

##### **'iv) Remove \$ symbols and compare**

**'\_\_\_\_\_**

```
SystemUtil.Run "C:\Program Files\Internet Explorer\iexplore.exe","","C:\Documents and Settings\gcr.GCRC-9A12FBD3D9","open"
Browser("Google").Page("Google").Sync
Browser("Google").Navigate "http://www.jjperfumes.com/"
```

```
Browser("Google").Page("JJ Perfumes-Discout perfume").Image("thumb_1845_WLINP50PSW").Click
Browser("Google").Page("JJ Perfumes-Discout perfume_2").WebEdit("quantity").Set "4"
Browser("Google").Page("JJ Perfumes-Discout perfume_2").WebButton("Add To Cart").Click
```

```
Unit_Price = Browser("Google").Page("JJ Perfumes-Discout perfume_3").WebElement("[Remove]").GetROProperty("innertext")
Qty = Browser("Google").Page("JJ Perfumes-Discout perfume_3").WebEdit("quantity").GetROProperty("value")
Cost = Browser("Google").Page("JJ Perfumes-Discout perfume_3").WebElement("[Remove_2]").GetROProperty("innertext")
```

```
Qty=Cint(Qty)
Unit_Price=Cdbl (Mid (Unit_Price,3))
Cost=Cdbl (Mid (Cost, 3))
'Msgbox Unit_Price: Msgbox Qty: Msgbox Cost
```

```
If Cost=Qty * Unit_Price Then
Reporter.ReportEvent micPass,"Res","Cost is Correct"
Else
Reporter.ReportEvent micFail,"Res","Cost is InCorrect"
End If
```

#### **Function To Create HTML Report**

```
Dim StartTime,stTime, enTime
```

```
Dim fso, ts,intCnt, intPass,intFail
```

```
intPass=0
```

```
intFail=0
```

```
Const ForWriting = 2
```

```
Function OpenFile (strFileName)
```

```
StartTime = Timer
```

```
stTime = Time
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
Set ts = fso.OpenTextFile(strFileName,2)
```

```
'OpenFile = strFileURL
```

CreateHeader

End Function

\*\*\*\*\*

#### **Function To Close File**

Dim objIE,strFileURL

Function CloseFile ( strFileURL,strEnv )

Footer(strEnv)

ts.close()

Set objIE = CreateObject("InternetExplorer.Application")

objIE.visible = True

objIE.Navigate strFileURL

wait(5)

Set objIE=nothing

End Function

\*\*\*\*\*

#### **Function to Clear Cookies()**

Function ClearCookies()

SystemUtil.Run "iexplore.exe"

WebUtil.DeleteCookies

Set objBrowser=Description.Create

objBrowser("micclass").value="Browser"

Set objBCount = Desktop.ChildObjects(objBrowser)

intBrowserCount = objBCount.Count

strHwnd=Browser("creationtime:=" & intBrowserCount).GetROProperty("hwnd")

SystemUtil.CloseProcessByHwnd(strHwnd)

End Function

\*\*\*\*\*

#### **Function to send Email**

Function SendMail()

If k=0 Then

'Nothing

Else

msgsub = " Geo\_PostalCode pattern searchTest Results:" &vbCrLf &"One or more of the Test Cases Failed." &vbCrLf &"See attachment for details."

End If

Set objConf = CreateObject("CDO.Configuration")

cdoSendUsingPort=2

sMailServerName="smtp.phx.move.com"

cdoAnonymous=cdoNONE

objConf.fields.item("http://schemas.microsoft.com/cdo/configuration/sendusing") = cdoSendUsingPort

objConf.fields.item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = sMailServerName

objConf.fields.item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate") = cdoAnonymous 'cdoBasic

objConf.fields.Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") = False

objConf.fields.Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 25

'objConf.fields.item("http://schemas.microsoft.com/cdo/configuration/sendusername") = "corp\ NRaoJ "

'objConf.fields.item("http://schemas.microsoft.com/cdo/configuration/sendpassword") = "123K!r45"

objConf.fields.Item("http://schemas.microsoft.com/cdo/configuration/smtpconnectiontimeout") = 60

objConf.fields.Update

Set objMsg = CreateObject("CDO.Message")

objMsg.Configuration = objConf

```
objMsg.to = "Naseer@gmail.com"

objMsg.From = "Naseer@gmail.com"

objMsg.Subject = "~Geo_PostalCode pattern search"

objMsg.TextBody=msgsub

doc=strFilepath

objMsg.AddAttachment(doc)

objMsg.Send

Set objMsg = Nothing

NewMail = True

End Function
```

\*\*\*\*\*

#### **Function to create append excel**

```
Function LogFile

Set objExcel=CreateObject("Excel.Application")

objExcel.visible=False

objExcel.workbooks.open(strFilePath)

r=1

Do Until len(objExcel.cells(r,1))=0

r=r+1

Loop

objExcel.cells(r,1).value="Step Name"

objExcel.cells(r,2).value="Step Description"

objExcel.cells(r,3).value="Test Results"
```

```
objExcel.DisplayAlerts = False
```

```
objExcel.Save
```

```
objExcel.Quit
```

```
Set objExcel = Nothing
```

```
End Function
```

```
*****
```

### **Function to get the Application Environment on which the script is running**

```
Function GetEnv()
```

```
Set fso=Createobject("Scripting.FileSystemObject")
```

```
sFile="C:\WINDOWS\system32\drivers\etc\hosts"
```

```
Set MyFile=fso.OpenTextFile(sFile,1, True)
```

```
Do
```

```
sEnv=MyFile.ReadLine
```

```
If instr(sEnv,"PRODUCTION")>0 Then
```

```
GetEnv="Production"
```

```
Exit do
```

```
Elseif instr(sEnv,"Staging")>0 Then
```

```
GetEnv="Staging"
```

```
Exit do
```

```
Elseif instr(sEnv,"QA MAINTENANCE")>0 Then
```

```
GetEnv="QAM"
```

```
Exit do
```

```
Elseif instr(sEnv,"qap.")>0 Then
```

```
GetEnv="qap"
```

Exit do

End If

Loop While MyFile.AtEndOfStream=False

End Function

\*\*\*\*\*

### Function To Create HTML Report

Dim StartTime,stTime, enTime

Dim fso, ts,intCnt, intPass,intFail

intPass=0

intFail=0

Const ForWriting = 2

Function OpenFile (strFileName)

StartTime = Timer

stTime = Time

Set fso = CreateObject("Scripting.FileSystemObject")

Set ts = fso.OpenTextFile(strFileName,2)

'OpenFile = strFileURL

CreateHeader

End Function

### Web Scripts-II

Web Application Testing using QTP.

Web forms verification, Object state verification, Database Testing and Links verification.

#### 1) Script to get the list of links in Google and do spell check

=====

```

Dim d
set mw=CreateObject("Word.Application")
set d=Description.Create
d("micclass").value="Link"
set a=Browser("Google").page("Google").childobjects(d)
for i=0 to a.count-1
mw.WordBasic.filenew
s=a(i).getROProperty("innertext")
mw.WordBasic.insert s
if mw.ActiveDocument.Spellingerrors.count>0 then
Reporter.ReportEvent 1,"Spelling","spelling error :"&s
end if
mw.ActiveDocument.Close(False)
next
mw.quit
set mw=nothing
2) Script to check ON the checkboxes in yahoo mail inbox
=====

```

```

Dim d

Set d=Description.Create

d("micclass").value="WebCheckBox"

Set c=Browser("Inbox (17) – Yahoo! Mail").Page("Inbox (17) – Yahoo! Mail").ChildObjects(d)

For i=1 to 10

c(i).set "ON"

Next
3) script to select a mail having subject 'hi' or 'HI'
=====

```

```

n=Browser("yahoo").Page("yahoo").WebTable("Inbox").RowCount
For i=2 to n
s=Browser("yahoo").Page("yahoo").WebTable("Inbox").GetCellData(i,7)
If lcase(trim(s))="hi" Then
Browser("yahoo").Page("yahoo").WebCheckBox("index:="&i-1).set "ON"
End If
Next

```

#### **4) Function to send a mail**

=====

```

Function SendMail(SendTo, Subject, Body, Attachment)

Set otl=CreateObject("Outlook.Application")

Set m=otl.CreateItem(0)

m.to=SendTo

m.Subject=Subject

m.Body=Body

If (Attachment <> "") Then

```



```
Mail.Attachments.Add(Attachment)
```

```
End If
```

```
m.Send
```

```
otl.Quit
```

```
Set m = Nothing
```

```
Set otl = Nothing
```

```
End Function
```

```
Call SendMail("nagesh.rao46@gmail.com","hi","This is test mail for teting","")
```

```
Adv VBScripts
```

#### **5) Open Internet Explorer and navigate to yahoo mail**

```
=====
```

```
Dim ie
```

```
Set ie=CreateObject("InternetExplorer.Application")
```

```
ie.Visible=True
```

```
ie.Navigate "www.yahoomail.com"
```

```
x=Browser("CreationTime:=0").GetROProperty("title")
```

```
msgbox x
```

#### **6) Function for Counting Objects from any opened web page**

```
Function Objects_Count(myObject)
```

```
Dim Objects
```

```
Set Objects=Description.Create
```

```
Objects("micclass").value=myObject
```

```
Set Object=Browser("title:=.*").Page("title:=.*").ChildObjects(Objects)
```

```
TotObjects=Object.Count
```

```
Msgbox TotObjects
```

```
End Function
```

```
Call Objects_Count("WebButton")
```

## 7) Create script for links validation and set results into an excel file

```
Set objDesc=Description.Create
```

```
objDesc("micclass").value="Link"
```

```
Set objColl=Browser("title:=.*").page("title:=.*").ChildObjects(objDesc)
```

```
msgbox objColl.count
```

```
Set objExcel=CreateObject("Excel.application")
```

```
objExcel.Visible=True
```

```
objExcel.Workbooks.Add
```

```
set objSheet=objExcel.ActiveSheet
```

```
objSheet.cells(1,1)="LinkName"
```

```
set c1=objSheet.cells(1,1)
```

```
c1.font.color=vbBlue
```

```
objSheet.cells(1,2)="TargetUrl"
```

```
Set c2=objSheet.cells(1,2)
```

```
c2.font.color=vbBlue
```

```
objSheet.cells(1,3)="ActualUrl"
```

```
Set c3=objSheet.cells(1,3)
```

```
c3.font.color=vbBlue
```

```
objSheet.cells(1,4)="Status"
```

```
Set c4=objSheet.cells(1,4)
```

```
c4.font.color=vbBlue
```

```
For i=0 to objColl.count-37 step 1
```

```
strName=Browser("title:=.*").page("title:=.*").Link("index:="&i).GetRoProperty("name")
```

```
TargetUrl=Browser("title:=.*").page("title:=.*").Link("index:="&i).GetRoProperty("url")
```

```
msgbox TargetUrl
```

```
Browser("title:=.*").page("title:=.*").Link("index:="&i).click
```

```
wait(4)
```

```
ActualUrl=Browser("title:=.*").GetRoProperty("url")
```

msgbox ActualUrl

If instr (1,TargetUrl,ActualUrl,1) > 0 Then

Reporter.ReportEvent micPass,"Link Name "&strName,"Link Validation done"

objSheet.cells(i+2,1)=strName

objsheet.cells(i+2,2)=TargetUrl

objsheet.cells(i+2,3)=ActualUrl

objsheet.cells(i+2,4)="Link Validation done"

Set c5= objsheet.cells(i+2,4)

c5.font.color=vbGreen

Else

Reporter.ReportEvent micFail,"Link Name "&strName,"Link validation fail"

objSheet.cells(i+2,1)=strName

objsheet.cells(i+2,2)=TargetUrl

objsheet.cells(i+2,3)=ActualUrl

objsheet.cells(i+2,4)="Link Validation fail"

Set c5= objsheet.cells(i+2,4)

c5.font.color=vbRed

End If

Browser("title:=.\*").Back

Next

Set objWbook=objExcel.ActiveWorkbook

objWbook.SaveAs "E:\Naseer.xls"

objExcel.Quit

Set objExcel=nothing

QTP Training

Error Handling in QTP

Error Handling:

Error handling refers to the anticipation, detection, and resolution of

programming, application, and communications errors.

Within every Script we have to think about possible exceptions and how to handle them. Especially in the uppermost layers of the script, it is important to handle all exceptions.

Error Handling in QuickTest Professional

QTP and VBScript give the Test Engineer some tools to handle errors and

Exceptions.

Error Preventing:

A good method for using error handling is to try to prevent them.

When an error occurred, Report it in detail. When working with GUI objects, use the Window.Exists property. Every If...Then..End If statement has the Else part, the same for Select Case. Use Case Else.

Error Handling Methods in QTP and VB Script:

- a) Synchronization
- b) Exist Property
- c) Recover Scenarios
- d) On Error Statement

Without an On Error statement, any run-time error that occurs is fatal: an error message is displayed, and execution stops.

Whenever possible, you use structured exception handling in your code, rather than resorting to unstructured exception handling and the On Error statement.

[www.Naseer.com](http://www.Naseer.com)

1

Parts:

GoToline

Enables the error-handling routine that starts at the line specified in the required line argument. The line argument is any line label or line number. If a run-time error occurs, control branches to the specified line, making the error handler active. The specified line must be in the same procedure as the On Error statement, or a compile-time error will occur.

GoTo 0

Disables enabled error handler in the current procedure and resets it to

Nothing.

GoTo -1

Disables enabled exception in the current procedure and resets it to

Nothing.

Resume Next

Specifies that when a run-time error occurs, control goes to the statement immediately following the statement where the error occurred, and execution continues from that point. Use this form rather than On Error GoTo when accessing objects.

on Error Resume Next

Dim a

a = 1

b2

MsgBox a + b

```
Function Sum( nNum1, nNum2)
If IsNumeric(nNum1) = False Or IsNumeric(nNum2) = False Then
On Error Resume Next
Err.Raise vbObjectError + 100, "Sum Function", _
"One or more parameters are invalid."
Exit Function
End If
Sum = nNum1 + nNum2
```

End Function

Call Sum(gcr,bannu)

[www.Naseer.com](http://www.Naseer.com)

QTP Training

e) Error Object

TheErr object is an intrinsic object with global scope — there is no need to create an instance of it in your code.

The properties of theErr object are set by the generator of an error — Visual

Basic, an Automation object, or the VBScript programmer.

The default property of theErr object isNumber.

Err.Number contains an integer and can be used by an Automation object

to return an SCODE.

When a run-time error occurs, the properties of the Err object are filled with information that uniquely identifies the error and information that can be used to handle it. To generate a run-time error in your code, use the Raise method.

The Err object's properties are reset to zero or zero-length strings ("" ) after an On Error Resume Next statement. The Clear method can be used to explicitly reset Err.

Example:

```
On Error Resume Next
Err.Raise 6 ' Raise an overflow error.
MsgBox ("Error # " & CStr(Err.Number) & " " & Err.Description)
Err.Clear
```

' Clear the error.

Err Object Properties and Methods

Properties

Description Property  
HelpContext Property  
HelpFile Property  
Number Property  
Source Property

Methods

Clear Method

Raise Method

[www.Naseer.com](http://www.Naseer.com)

3

## ) Exit Statement

**Open 30 to 40 Orders in Flight Reservation Window, if some records not available handle the situation**

```
Option Explicit
Dim Order_Number
If Not Window("Flight Reservation").Exist(3) Then
SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe", "", "C:\Program Files\HP\QuickTest Professional\samples\flight\app\","open"
Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set "Naseer"
Dialog("Login").WinEdit("Password:").SetSecure "4c2e1e65bf29943393b6940f116d35231ce5fb7e"
Dialog("Login").WinButton("OK").Click
End If
```

```
For Order_Number= 30 to 40 step 1
Window("Flight Reservation").Activate
Window("Flight Reservation").WinButton("Button").Click
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set Order_Number
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click
```

```
If Window("Flight Reservation").Dialog("Open Order").Dialog("Flight Reservations").Exist(3) Then
Window("Flight Reservation").Dialog("Open Order").Dialog("Flight Reservations").WinButton("OK").Click
Window("Flight Reservation").Dialog("Open Order").WinButton("Cancel").Click
Reporter.ReportEvent micWarning,"Res", "Up to "& Order_Number-1 &" Order only available"
Exit For
End If
Next
```

## QTP Interview Questions Part-V

### 1) What are the extensions of QTP files?

- o QTP Test / Test Script(.mts){.mts=module/mercury test script}
- o Local Repository file (.mtr){.mtr=module/mercury test repository}
- o Shared repository files(.tsr){Test shared repository}
- o Library files (.vbs)
- o Environmental files (.xml)
- o Test batch file (.mtb){module test batch}
- o QTP result file (xml) optional: pdf, doc, xml-for reporting
- o Recovery scenario file (.qrs){quick recovery scenario}

### 2) How to handle dynamic objects?

By using Regular expressions we can handle dynamic objects.

### 3) What are the differences between checkpoint and output value?

Both are same in insertion but different in purpose.

Checkpoint is a verification point, it takes expected results from the user and compare with the actual results during execution and provides test results.

Output value captures object properties values, database content and xml elements values and stores in run time data table.

Checkpoints compare actual results with expected results, output values captures values and stores.

#### **4) How to handle Repositories dynamically?**

With Repositoriescollection utility object, we can handle shared object repositories dynamically.

Note: on Repositoriescollection object, we can use Add, Find, Remove, and Removeall like methods.

#### **5) What is the use of environment variables?**

Environment variables are global variables that we can use from number of tests commonly.

Whenever we want to use some variables from number of tests, instead of declaring as local variables, if we use environment variables concept then maintenance will be easy.

#### **6) What is a Regular Expression, where we use this feature in QTP?**

Regular expression is nothing but matching patterns.

In QTP, we use this feature for identifying dynamic objects

#### **7) What is the advantage of Dynamic Descriptive Programming than Static?**

Incase of dynamic programming Tests generation is easy and maintenance also very easy.

#### **8) What are the limitations of QuickTest?**

- a. It supports MS Windows operating environment only
- b. It doesn't support CLI Applications (Command line interface)
- c. It supports IE only for generating scripts but it does not support Mozilla and other browsers. For executing tests it supports Mozilla like other browsers.
- d. It has limited support for web services testing.



**9) How to capture a moving image in QTP?**

**10) What is Smart Identification in QTP and explain it?**

**11) How many folders are created in one Test creation in QTP?**

Two folders and six files.

**12) What is Description object in QTP?**

**13) What are the Properties you use for identifying a Browser, a Page when using descriptive programming?**

title property for Browser

title property for Page

**14) How do you measure Transaction Time in QTP other than Using of Start and End Transaction Points?**

By using '**Timer**' function we can measure transaction times.

**15) what is the Deference between Recovery Scenario and "On Error Resume Next"?**

1) Both are for handling Error events, Recovery Scenario is a QTP Tool feature, whereas "On Error Resume Next" is a VB Script feature

2) "On Error Resume Next" is used when we are sure about the step where the error can occur. Recovery Scenario can be used to handle unexpected event or error.

\*\*\*\*\*

**On Error Statement (Visual Basic)**

Enables an error-handling routine and specifies the location of the routine within a procedure; can also be used to disable an error-handling routine.

Without an **On Error** statement, any run-time error that occurs is fatal: an error message is displayed, and execution stops.

Whenever possible, we suggest you use structured exception handling in your code, rather than resorting to unstructured exception handling and the **On Error** statement.

#### 16) Explain in brief about the QTP Automation Object Model.

QTP Automation Object model **deals with Automation of QTP itself**. Almost all configuration and functionality provided by QTP is represented by QTP's Automation Object Model . Almost all dialog boxes in QTP have a corresponding automation object which can set or retrieved using the corresponding properties or methods in the Automation Object Model.QTP Automation Objects can be used along with standard VB programming elements like iterative loops or conditional statements to help you design a script of choice

#### 17) What is Test Fusion Report ?

Test Fusion Report , displays all aspects of a test run and is organized in a **Tree format**.

It gives details of each step executed for all iterations.

It also gives Run-time data table, Screen shots and movie of the test run if opted.

#### 18) How can you handle exceptions in QTP?

In QTP Exceptional handling is done by using

- a. Recovery Scenarios.
- b. Using "On Error" statement

In Recovery scenario you have to define.

1. Triggered Events.
2. Recovery steps.
3. Post Recovery Test-Run.

At Script Level you can use the On Error Resume Next and On Error Go to 0 statement.

#### 19) What is the Difference between Bitmap Check point & Image Check point?

**Bitmap** checkpoint does a **pixel to pixel comparison** of an image or part of an image.

**Image** checkpoint does do a pixel to pixel comparison but instead **compare image properties** like alt text , destination url etc.

**20) How would you export a Script from one PC to another in QTP ?**

We can make use of the **“Generate Script”** function available in Object Identification, Test Settings and Tools/Options tab to create a zip of the script at the source computer. These zip files then can be imported into QTP at the destination computer.

**21) Can launch two instances of QTP on the same machine ?**

**No.** You can work with only single instance of QTP on the same machine. But QTP itself can work on multiple instances of the Application Under Test (AUT). Ex: QTP can handle multiple IE browser windows.

**22) What is SetToProperty ?**

SetToProperty changes property of an object stored in the Object Repository. However these changes are not permanent.

**23) What is Accessibility Checkpoint?**

World Wide Web Consortium (W3C) came up with some **instructions and guidelines** for Web-based technology and information systems **to make it easy for the disabled to access the web**. For example the standards make it mandatory to have an 'alt text' for an image. So a blind person who is accessing the website, will use text – to –speech converters and atleast understand what the image is about if not see it. All these standards are checked by Accessibility Checkpoints.

**24) What is the difference between Properties and methods?**

A property is a named attribute of an object. Properties define the characteristics of an object such as Size, Color etc. or sometimes the way in which it behaves.

A method is an action that can be performed on objects.

For example, a dog is an object. Its properties might include long white hair, blue eyes, 3 pounds weight etc. Its methods might be move, jump, play, breath etc.

## QTP Interview Questions Part-IV

**1) What are the differences between Actions and Functions and which are better?**

**Actions:**

- o It's a QTP feature and Local feature (not a global feature)
- o We can use check points in Actions
- o As it is QTP feature it may corrupt
- o Actions are slow execution

**Functions:**

- o It's a VB script feature and global concept
- o We can't use checkpoints in functions, instead of checkpoints we can use validation points (Loops and Conditional statements)
- o It doesn't corrupt
- o Functions are faster in execution than Actions

**2) How create methods in QTP?**

By associating procedures to test objects.

**3) How to enhance tests with windows API functions?**

With the help of WSH (It is a Windows administration tool), we can use Windows API functions for enhancing Tests.

**4) What are the major differences between functions and actions in QTP? What advantages there in functions than actions?**

Read 21st question and answer.

**5) What is Automation Object Model?**

It's a process of Automating the Quick test operations. To do this we can use java script, VC++ and .NET Apart from VBscript.

**6) What is Windows Script Host?**

WSH is a windows administration tool, it provides hosting environment for running VBscript, Jscript scripts.

**7) What is exception handling?**

Handling expected and unexpected errors during execution

**8) What is the difference between constants and variables in VB Script?**

We use constants for replacing literal values and constants never change.  
We use variables for storing values that returned by the program, functions, etc.  
Variable values may vary through out the script.

**9) How test engineer identifies when programmatic descriptions are useful?**

- o If application is not ready (using screenshots and documents we can prepare descriptive programming)
- o If we want to make our scripts global
- o If lengthy tests are there and that we want to execute fast

**10) How to retrieve data from application objects?**

Using loop statements and GetROProperty method

**11) How to build scripts that access data from external resources?**

We can fetch test data directly from external files and databases by creating automation objects  
Ex:ADODB, Filesystemobject

**12) What is object spy used for?**

It is a feature of QTP, used for getting the object's information (all properties and values) .It is available in QTP tool main window, in Local repository and in object Repository Manager

**13) What is the purpose of input parameters?**

For Passing values

**14) What is an Action? How many types of Actions are there in QTP?**

Set of statements to perform a task or tasks is called an Action.

Three types of actions are there.

1. non reusable actions
2. reusable actions
3. External actions

In QTP 9.2 default action is: Non reusable

In QTP 10.0 default action is: Reusable

Example:

Let there are two tests : test:1 and test:2

In test:1 let there are two actions

a.login (reusable)and b. insert order(nonreusable)

in test:2 let there are two actions

a.open order(reusable) and b. fax order(non reusable)

If we call login action from test:1 in test :2 then login action is called External action.

In test :2 login is reusable but calling point of view its external action

#### **15) What is a virtual object? How to configure virtual objects?**

Whenever QTP is not recognizing an area or user defined object as an object that we can make as a virtual object.

For making virtual objects QTP is providing a feature called Virtual object manager. Through this we can configure virtual objects.

#### **16) What are dynamic objects?**

The object, which properties change dynamically (frequently). We can use Regular Expressions to handle dynamic objects.

#### **17) What is the purpose of description object?**

The purpose of description object is to store multiple properties values of a test object.

#### **18) What is the advantage of associating a procedure to a test object?**

By associating a procedure to a test object we can use this as a method on that test object.

#### **19) How to create a Library File?**

Open an Editor (Ex: Notepad),

Generate some script statements and save with .vbs extension.

In a library file we can put more than one function also

#### **20) What are the disadvantages of using actions?**

- o Action is a QTP tool feature that occupies QTP's more memory space.
- o It may be corrupted.
- o It is not a global feature. Test engineer may not have the control on action.

#### **21) What is the difference between Test Objects and Run Time Objects ?**

Test objects are basic and generic objects that QTP recognize. Run time object means the actual object to which a test object maps.

#### **22) Can i change properties of a test object?**

Yes. You can use SetTOProperty to change the test object properties. It is recommended that you switch off the Smart Identification for the

object on which you use SetTOProperty function.

Can i change properties of a run time object?

No (but Yes also). You can use GetROProperty("outerText") to get the outerText of a object but there is no function like SetROProperty to change this property. But you can use WebElement().object.outerText="Something" to change the property.

#### **23) When and Why to use Descriptive programming?**

Below are some of the situations when Descriptive Programming can be considered useful:

The objects in the application are dynamic in nature and need special handling to identify the object. The best example would be of clicking a link which changes according to the user of the application, Ex. "Logout <<UserName>>".

When object repository is getting huge due to the no. of objects being added. If the size of Object repository increases too much then it decreases the performance of QTP while recognizing a object.

When you don't want to use object repository at all. Well the first question would be why not Object repository? Consider the following scenario which would help understand why not Object repository

Scenario 1: Suppose we have a web application that has not been developed yet. Now QTP for recording the script and adding the objects to repository needs the application to be up, that would mean waiting for the application to be deployed before we can start of with making QTP scripts. But if we know the descriptions of the objects that will be created then we can still start off with the script writing for testing

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 on any page.

Modification to a test case is needed but the Object repository for the same is Read only or in shared mode i.e. changes may affect other scripts as well. When you want to take action on similar type of object i.e. suppose we have 20 textboxes on the page and there 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.

## QTP Interview Questions Part-III

### Questions & Answers

#### 1) Explain about your Company's Automation Framework?

#### 2) What are the important factors in test tool selection?

a. Technical factors:

i. Nature of the AUT (Application Under Test)

ii. Tool Features (Facilities available)

iii. Availability of Skilled professionals

iv. Organizations experience

v. Technical support from vendor

b. Business factors:

i. Budget

ii. Vendor's Reputation

iii. Client's Interest

#### 3) What test cases can be automated?

Test cases that we want to execute on number of versions of the software

Ex: Regression test cases

Test cases that we want to execute with multiple sets of data

Ex: Data driven test cases

Test cases having complex functionality

#### **4) What test cases cannot be automated?**

- o Dynamically changing functionality
- o Test cases that require more Human interaction
- o CUI (command user interface) based test cases

#### **5) Explain about the Testing process in QTP?**

Planning:

- o Analyzing the Application
- o Automation Framework Implementation
- o Selecting/Generating Test cases for Automation
- o Collecting Test Data
- o Configuring Tool Settings

Generating the basic tests

Using Object Repository method or using Descriptive Programming, we can generate Tests

Enhancing Tests

With respect to our project requirements, we have to enhance our tests. For enhancing tests QTP is providing various features like:

- o Inserting Transaction points
- o Synchronization
- o Parameterization
- o Inserting VB script Flow Control statements
- o Adding comments
- o Calling Functions/ calling Actions
- o Using Environment variables
- o Entering Regular Expressions
- o Inserting Checkpoints
- o Inserting Output values

Etc..

Note: For any Test all enhancement methods not required.

#### Debugging Tests (if required)

It is optional; basically scripting languages are Light weight languages, no need to compile them separately. During execution Scripts can be compiled and Run.

If we have any doubts about some scripts, then we can execute them step by step with the support of VBScript Debug commands and Break points

#### Running tests

Generally, many companies are using Framework feature for Running Tests.

QTP Supports Individual Test execution, Batch execution, Tests execution through AOM Scripting, and Scheduled execution with the support of Windows Operating system.

#### Analyzing Results

It is not the Responsibility of QTP; Test engineer has to do this manually.

QTP is providing Result window in XML format.

QTP is providing Result deletion tool also.

QTP is providing Test results exporting facility. It supports HTML, Word document and PDF (portable document format) formats.

#### Reporting defects

Reporting defects also not a QTP responsibility that we have to do manually after analyzing the results, but if we are using Quality Center, from QTP result window we can send defects directly.

### **6) Explain about the object identification configuration process in QuickTest?**

### **7) What types of License available for QTP?**

a) Seat License or Node locked:

Fixed for one System (Machine), we can use from that System only



b) Concurrent License or Floating License:

This License, we can use from different Systems but one at a time

## **8) How to generate Basic scripts/tests from Manual Test cases in QTP?**

For generating basic tests QTP is providing 1) Object repository 2) Descriptive Programming

a) Object repository:

Recording (Local Repository)

And/or Shared object Repository

b) Descriptive programming or Programmatic Descriptions

## **9) What is the difference between Local and Shared object Repository?**

Local: QTP creates a Local Repository for every Action during Recording automatically

It can't be shared among other tests

It's a QTP internal file

Users no need to maintain separately (auto save)

Extension is .mtr – (mercury test repository/module test repository)

Shared: User creates shared repository by adding objects

That can be shared among number of tests

User has to maintain shared object repository files

Extension is .tsr – (test shared repository)

## **10) How to maintain objects in Object Repository?**

Same as above 9th

### **11) What is Synchronization?**

It is a process of matching the speeds of both QTP and AUT in order to get proper execution and results

### **12) How to synchronize QTP and AUT?**

For synchronizing QTP and AUT, QTP is providing below methods (features)

- o Wait statements
- o Synchronizations points
- o Increasing tool default synchronization time (Increase timeout)
- o Sync method (Only for web)
- o Using Exit property also we can use (GUI and web) – Limited usage

### **13) How to parametrize tests?**

It is a process of passing parameters for replacing constant values. In QTP several methods are available

a. Through Looping : Here some limitations are there :

We can use Looping for sequential numbers and logical numbers only, if it is a string we can't use.

- b. Through Data table
- c. Fetching test data directly from external files (excel, flat files)
- d. Fetching test data directly from databases
- e. Fetching data from AUT objects (front-end objects)

### **14) How to Generate Non-recordable steps?**

Using Step Generator, we can generate non recordable steps

**15) What are the benefits of Step Generator?**

It's a library of functions and utility objects. Through this we can generate recordable and non recordable steps, utility statements and built in functions.

**16) How to count Links in a web page without using page checkpoint?**

Using Description object and Childobjects method

**17) How to count all available objects in a dialog box /a window/a web page?**

Using Childobjects method

**18) What are the major disadvantages in record tests?**

Recording disadvantages:

- o It occupies a lot of memory space due to this, QTP performance will be reduced
- o Modifications are very difficult
- o User may not have command on the script, if any modifications are there user may not understand properly
- o Recorded scripts are QTP internal files, they may corrupt.

**19) What are the disadvantages in using of checkpoints?**

- o They are QTP internal files we can't use in external functions
- o As they are QTP internal files they may corrupt
- o As it is ready made feature, that has limited usage

**20) In QuickTest Pro can we insert Checkpoints programmatically? How?**

Yes. We can insert checkpoints programmatically with the help of GetRoProperty method.

## **21) What are Differences between Design time Data Table and Run-time Data Table?**

### **Design Time Data Table:**

It is viewed in the QTP Main Test

It is created prior to the test execution

It represents data from external Sources

### **Run-Time Data Table:**

It is viewed in the QTP Test Result Window

It is created in Test Results after test execution.

It represents a live version of design Time Data table.

### **Our QTP Point of view below are the learning objectives:**

- a) Adding Comments
- b) Data types
- c) Declarations (Variables (Scalar and Array), Constants)
- d) VB Script Operators
  - i) Arithmetic Operators (Including Concatination operators)
  - ii) Comparison
  - iii) Logical
- e) Flow Control Statements (a. Conditional Statements)
  - i) If...Then...Else...End If
  - ii) Select Case...Case...Case Else...End Select
- f) Flow Control Statements (b. Loop Statements)
  - i) For...Next
  - ii) While...Wend

- iii) Do While/Until...Loop
- iv) For Each...Next

- g) VB Script Procedures(Functions)
  - i) Built-in Functions(String, Array, Math, Date & Time, Conversion functions etc...)
  - ii) User defined
    - 1) Sub Procedures
    - 2) Function Procedures

- h) Coding Conventions
- 

- i) File System Operations

- j) Excel sheet Operations

- k) database Operations

- l) Other VB Script Objects
  - i) Dictionary Object
  - ii) Word
  - iii) Internet Explorer Object
  - iv) RegExp Object

- m) Regular expressions

- k) Error Handling

### **VB Script Variables**

Definition 1):

Variable is a named memory location for storing program information

Definition 2):

A variable is a convenient placeholder that refers to a computer memory location where we can store program information that may change during the time our script is running.

#### **Purpose of Variable:**

- a) Comparing values

Example:

```
Dim x,y,a
```

```
x=100
```

```
y=100
```

```
a=x=y
```

Msgbox a 'It returns True

b) Holding Program Result

Example:

Cost=Tickets\*Price

c) Passing parameters

d) To store data that returned by functions

Example:

myDate=Now ' It returns current data & time

e) To hold data

Example:

myName="Naseer"

### **Declaring Variables**

We declare variables explicitly in our script using the Dim statement, the Public statement, and the Private statement.

For example:

Dim city

Dim x

We declare multiple variables by separating each variable name with a comma. For

Example:

Dim x, y, city, Naseer

We can also declare a variable implicitly by simply using its name in our script. That is not generally a good practice because we could misspell the variable name in one or more places, causing unexpected results when our script is run. For that reason, the Option Explicit statement is available to require explicit declaration of all variables.

The Option Explicit statement should be the first statement in our script.

### **Option Explicit Statement**

Forces explicit declaration of all variables in a script.

Option Explicit ' Force explicit variable declaration.

Dim MyVar ' Declare variable.

MyInt = 10 ' Undeclared variable generates error.

MyVar = 10 ' Declared variable does not generate error.

### **Naming Restrictions for Variables**

Variable names follow the standard rules for naming anything in VBScript. A variable name:

a) Must begin with an alphabetic character.

Dim abc 'Right

Dim 9ab 'Wrong

Dim ab9 'Right

b) Cannot contain an embedded period.

Dim abc 'Right

Dim ab.c 'wrong

Dim ab-c 'wrong

Dim ab c 'wrong

Dim ab\_c 'Right

c) Must not exceed 255 characters.

d) Must be unique in the scope in which it is declared.

### **Scope of Variables**

A variable's scope is determined by where we declare it.

When we declare a variable within a procedure, only code within that procedure can access or change the value of that variable.

If we declare a variable outside a procedure, we make it recognizable to all the procedures in our script. This is a script-level variable, and it has script-level scope.

Example:

Dim x,y,z

x=10

y=20

z=x+y

msgbox z 'Returns 30

Function res

Dim a,b,c

a=30

b=40

c=a+b+y

msgbox c ' Returns 90

End Function

Call res

### **Life Time of Variables**

The lifetime of a variable depends on how long it exists.

The lifetime of a script-level variable extends from the time it is declared until the time the script is finished running.

At procedure level, a variable exists only as long as you are in the procedure.

### **Assigning Values to Variables**

Values are assigned to variables creating an expression as follows:

The variable is on the left side of the expression and the value you want to assign to the variable is on the right.

For example:

A = 200

City = "Hyderabad"

X=100; Y=200

### **Scalar Variables and Array Variables**

A variable containing a single value is a scalar variable.

A variable containing a series of values, is called an array variable.

Array variables and scalar variables are declared in the same way, except that the declaration of an array variable uses parentheses () following the variable name.

Example:

Dim A(3)

Although the number shown in the parentheses is 3, all arrays in VBScript are zero-based, so this array actually contains 4 elements.

We assign data to each of the elements of the array using an index into the array.

Beginning at zero and ending at 4, data can be assigned to the elements of an array as follows:

A(0) = 256

A(1) = 324

A(2) = 100

A(3) = 55

Similarly, the data can be retrieved from any element using an index into the particular array element you want.

For example:

SomeVariable = A(4)

Arrays aren't limited to a single dimension. We can have as many as 60 dimensions, although most people can't comprehend more than three or four dimensions.

In the following example, the MyTable variable is a two-dimensional array consisting of 6 rows and 11 columns:

Dim MyTable(5, 10)

In a two-dimensional array, the first number is always the number of rows; the second number is the number of columns.



### Dynamic Arrays

We can also declare an array whose size changes during the time our script is running. This is called a dynamic array.

The array is initially declared within a procedure using either the Dim statement or using the ReDim statement.

However, for a dynamic array, no size or number of dimensions is placed inside the parentheses.

For example:

```
Dim MyArray()  
ReDim AnotherArray()
```

To use a dynamic array, you must subsequently use ReDim to determine the number of dimensions and the size of each dimension.

In the following example, ReDim sets the initial size of the dynamic array to 25. A subsequent ReDim statement resizes the array to 30, but uses the Preserve keyword to preserve the contents of the array as the resizing takes place.

```
ReDim MyArray(25)
```

```
ReDim Preserve MyArray(30)
```

There is no limit to the number of times we can resize a dynamic array, although if we make an array smaller, we lose the data in the eliminated

### SQL Queries

Create the following Tables:

LOCATION	
Location_ID	Regional_Group
122	NEW YORK
123	DALLAS
124	CHICAGO
167	BOSTON

DEPARTMENT		
Department_ID	Name	Location_ID
10	ACCOUNTING	122
20	RESEARCH	124
30	SALES	123
40	OPERATIONS	167

JOB	
Job_ID	Function
667	CLERK
668	STAFF
669	ANALYST
670	SALESPERSON
671	MANAGER
672	PRESIDENT

EMPLOYEE									
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	MIDDLE_NAME	JOB_ID	MANAGER_ID	HIRE DATE	SALARY	COMM	DEPARTMENT_ID
7369	SMITH	JOHN	Q	667	7902	17-DEC-84	800	NULL	20
7499	ALLEN	KEVIN	J	670	7698	20-FEB-85	1600	300	30
7505	DOYLE	JEAN	K	671	7839	04-APR-85	2850	NULL	30
7506	DENNIS	LYNN	S	671	7839	15-MAY-85	2750	NULL	30
7507	BAKER	LESLIE	D	671	7839	10-JUN-85	2200	NULL	40
7521	WARK	CYNTHIA	D	670	7698	22-FEB-85	1250	500	30

**Queries based on the above tables:**

**Simple Queries:**

1. List all the employee details
2. List all the department details
3. List all job details
4. List all the locations
5. List out first name,last name,salary, commission for all employees
6. List out employee\_id,last name,department id for all employees and rename employee id as "ID of the employee", last name as "Name of the employee", department id as "department ID"
7. List out the employees annual salary with their names only.

**Where Conditions:**

1. List the details about "SMITH"
2. List out the employees who are working in department 20
3. List out the employees who are earning salary between 3000 and 4500
4. List out the employees who are working in department 10 or 20
5. Find out the employees who are not working in department 10 or 30
6. List out the employees whose name starts with "S"
7. List out the employees whose name start with "S" and end with "H"
8. List out the employees whose name length is 4 and start with "S"
9. List out the employees who are working in department 10 and draw the salaries more than 3500
10. list out the employees who are not receiving commission.

**Order By Clause:**

1. List out the employee id, last name in ascending order based on the employee id.
2. List out the employee id, name in descending order based on salary column
3. list out the employee details according to their last\_name in ascending order and salaries in descending order
4. list out the employee details according to their last\_name in ascending order and then on department\_id in descending order.

### **Group By & Having Clause:**

1. How many employees who are working in different departments wise in the organization
2. List out the department wise maximum salary, minimum salary, average salary of the employees
3. List out the job wise maximum salary, minimum salary, average salaries of the employees.
4. List out the no.of employees joined in every month in ascending order.
5. List out the no.of employees for each month and year, in the ascending order based on the year, month.
6. List out the department id having atleast four employees.
7. How many employees in January month.
8. How many employees who are joined in January or September month.
9. How many employees who are joined in 1985.
10. How many employees joined each month in 1985.
11. How many employees who are joined in March 1985.
12. Which is the department id, having greater than or equal to 3 employees joined in April 1985.

### **Sub-Queries**

1. Display the employee who got the maximum salary.
2. Display the employees who are working in Sales department
3. Display the employees who are working as "Clerk".
4. Display the employees who are working in "New York"
5. Find out no.of employees working in "Sales" department.
6. Update the employees salaries, who are working as Clerk on the basis of 10%.
7. Delete the employees who are working in accounting department.
8. Display the second highest salary drawing employee details.
9. Display the Nth highest salary drawing employee details

### **Sub-Query operators: (ALL,ANY,SOME,EXISTS)**

1. List out the employees who earn more than every employee in department 30.
2. List out the employees who earn more than the lowest salary in department 30.
3. Find out whose department has not employees.
4. Find out which department does not have any employees.

### **Co-Related Sub Queries:**

47. Find out the employees who earn greater than the average salary for their department.

## *Joins*

### **Simple join**

48. List our employees with their department names

49. Display employees with their designations (jobs)

50. Display the employees with their department name and regional groups.

51. How many employees who are working in different departments and display with department name.

52. How many employees who are working in sales department.

53. Which is the department having greater than or equal to 5 employees and display the department names in ascending order.

54. How many jobs in the organization with designations.

55. How many employees working in "New York".

### **Non – Equi Join:**

56. Display employee details with salary grades.

57. List out the no. of employees on grade wise.

58. Display the employee salary grades and no. of employees between 2000 to 5000 range of salary.

### **Self Join:**

59. Display the employee details with their manager names.

60. Display the employee details who earn more than their managers salaries.

61. Show the no. of employees working under every manager.

**Outer Join:**

61.Display employee details with all departments.

62.Display all employees in sales or operation departments.

**Set Operators:**

63.List out the distinct jobs in Sales and Accounting Departments.

64.List out the ALL jobs in Sales and Accounting Departments.

65.List out the common jobs in Research and Accounting Departments in ascending order.

**Answers**

1. SQL > Select \* from employee;
2. SQL > Select \* from department;
3. SQL > Select \* from job;
4. SQL > Select \* from loc;
5. SQL > Select first\_name, last\_name, salary, commission from employee;
6. SQL > Select employee\_id "id of the employee", last\_name "name", department\_id as "department id" from employee;
7. SQL > Select last\_name, salary\*12 "annual salary" from employee
8. SQL > Select \* from employee where last\_name='SMITH';
9. SQL > Select \* from employee where department\_id=20
10. SQL > Select \* from employee where salary between 3000 and 4500
11. SQL > Select \* from employee where department\_id in (20,30)
12. SQL > Select last\_name, salary, commission, department\_id from employee where department\_id not in (10,30)
13. SQL > Select \* from employee where last\_name like 'S%'
14. SQL > Select \* from employee where last\_name like 'S%H'
15. SQL > Select \* from employee where last\_name like 'S\_\_'
16. SQL > Select \* from employee where department\_id=10 and salary>3500
17. SQL > Select \* from employee where commission is Null
18. SQL > Select employee\_id, last\_name from employee order by employee\_id
19. SQL > Select employee\_id, last\_name, salary from employee order by salary desc
20. SQL > Select employee\_id, last\_name, salary from employee order by last\_name, salary desc
21. SQL > Select employee\_id, last\_name, salary from employee order by last\_name, department\_id desc

22. SQL > Select department\_id, count(\*), from employee group by department\_id
23. SQL > Select department\_id, count(\*), max(salary), min(salary), avg(salary) from employee group by department\_id
24. SQL > Select job\_id, count(\*), max(salary), min(salary), avg(salary) from employee group by job\_id
25. SQL > Select to\_char(hire\_date,'month')month, count(\*) from employee group by to\_char(hire\_date,'month') order by month
26. SQL > Select to\_char(hire\_date,'yyyy') Year, to\_char(hire\_date,'mon') Month, count(\*) "No. of employees" from employee group by to\_char(hire\_date,'yyyy'), to\_char(hire\_date,'mon')
27. SQL > Select department\_id, count(\*) from employee group by department\_id having count(\*)>=4
28. SQL > Select to\_char(hire\_date,'mon') month, count(\*) from employee group by to\_char(hire\_date,'mon') having to\_char(hire\_date,'mon')='jan'
29. SQL > Select to\_char(hire\_date,'mon') month, count(\*) from employee group by to\_char(hire\_date,'mon') having to\_char(hire\_date,'mon') in ('jan','sep')
30. SQL > Select to\_char(hire\_date,'yyyy') Year, count(\*) from employee group by to\_char(hire\_date,'yyyy') having to\_char(hire\_date,'yyyy')=1985
31. SQL > Select to\_char(hire\_date,'yyyy')Year, to\_char(hire\_date,'mon') Month, count(\*) "No. of employees" from employee where to\_char(hire\_date,'yyyy')=1985 group by to\_char(hire\_date,'yyyy'),to\_char(hire\_date,'mon')
32. SQL > Select to\_char(hire\_date,'yyyy')Year, to\_char(hire\_date,'mon') Month, count(\*) "No. of employees" from employee where to\_char(hire\_date,'yyyy')=1985 and to\_char(hire\_date,'mon')='mar' group by to\_char(hire\_date,'yyyy'),to\_char(hire\_date,'mon')
33. SQL > Select department\_id, count(\*) "No. of employees" from employee where to\_char(hire\_date,'yyyy')=1985 and to\_char(hire\_date,'mon')='apr' group by to\_char(hire\_date,'yyyy'), to\_char(hire\_date,'mon'), department\_id having count(\*)>=3
34. SQL > Select \* from employee where salary=(select max(salary) from employee)
35. SQL > Select \* from employee where department\_id IN (select department\_id from department where name='SALES')
36. SQL > Select \* from employee where job\_id in (select job\_id from job where function='CLERK')
37. SQL > Select \* from employee where department\_id=(select department\_id from department where location\_id=(select location\_id from location where regional\_group='New York'))
38. SQL > Select \* from employee where department\_id=(select department\_id from department where name='SALES' group by department\_id)
39. SQL > Update employee set salary=salary\*10/100 where job\_id=(select job\_id from job where function='CLERK')
40. SQL > delete from employee where department\_id=(select department\_id from department where name='ACCOUNTING')
41. SQL > Select \* from employee where salary=(select max(salary) from employee where salary <(select max(salary) from employee))
42. SQL > Select distinct e.salary from employee where & no-1=(select count(distinct salary) from employee where sal>e.salary)
43. SQL > Select \* from employee where salary > all (Select salary from employee where department\_id=30)
44. SQL > Select \* from employee where salary > any (Select salary from employee where department\_id=30)
45. SQL > Select employee\_id, last\_name, department\_id from employee e where not exists (select department\_id from department d where d.department\_id=e.department\_id)
46. SQL > Select name from department d where not exists (select last\_name from employee e where d.department\_id=e.department\_id)
47. SQL > Select employee\_id, last\_name, salary, department\_id from employee e where salary > (select avg(salary) from employee where department\_id=e.department\_id)
48. SQL > Select employee\_id, last\_name, name from employee e, department d where e.department\_id=d.department\_id
49. SQL > Select employee\_id, last\_name, function from employee e, job j where e.job\_id=j.job\_id
50. SQL > Select employee\_id, last\_name, name, regional\_group from employee e, department d, location l where e.department\_id=d.department\_id and d.location\_id=l.location\_id
51. SQL > Select name, count(\*) from employee e, department d where d.department\_id=e.department\_id group by name
52. SQL > Select name, count(\*) from employee e, department d where d.department\_id=e.department\_id group by name having name='SALES'
53. SQL > Select name, count(\*) from employee e, department d where d.department\_id=e.department\_id group by name having count (\*)>=5 order by name
54. SQL > Select function, count(\*) from employee e, job j where j.job\_id=e.job\_id group by function
55. SQL > Select regional\_group, count(\*) from employee e, department d, location l where e.department\_id=d.department\_id and d.location\_id=l.location\_id and regional\_group='NEW YORK' group by regional\_group
56. SQL > Select employee\_id, last\_name, grade\_id from employee e, salary\_grade s where salary between lower\_bound and upper\_bound order by last\_name
57. SQL > Select grade\_id, count(\*) from employee e, salary\_grade s where salary between lower\_bound and upper\_bound group by grade\_id order by grade\_id desc
58. SQL > Select grade\_id, count(\*) from employee e, salary\_grade s where salary between lower\_bound and upper\_bound and lower\_bound>=2000 and lower\_bound<=5000 group by grade\_id order by grade\_id desc
59. SQL > Select e.last\_name emp\_name, m.last\_name, mgr\_name from employee e, employee m where e.manager\_id=m.employee\_id
60. SQL > Select e.last\_name emp\_name, e.salary emp\_salary, m.last\_name, mgr\_name, m.salary mgr\_salary from employee e, employee m where e.manager\_id=m.employee\_id and m.salary<e.salary
61. SQL > Select m.manager\_id, count(\*) from employee e, employee m where e.employee\_id=m.manager\_id group by m.manager\_id
62. SQL > Select last\_name, d.department\_id, d.name from employee e, department d where e.department\_id(+) =d.department\_id
63. SQL > Select last\_name, d.department\_id, d.name from employee e, department d where e.department\_id(+) =d.department\_id and d.department\_id in (select department\_id from department where name IN ('SALES','OPERATIONS'))

64. SQL > Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='SALES')) union Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='ACCOUNTING'))
65. SQL > Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='SALES')) union all Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='ACCOUNTING'))
66. SQL > Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='RESEARCH')) intersect Select function from job where job\_id in (Select job\_id from employee where department\_id=(select department\_id from department where name='ACCOUNTING')) order by function

## **Manual Testing Questions and Answers**

### **1. What is Acceptance Testing?**

Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

### **2. What is Accessibility Testing?**

Verifying a product is accessible to the people having disabilities (deaf, blind, mentally disabled etc.).

### **3. What is Ad Hoc Testing?**

A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality. Can include negative testing as well. See also Monkey Testing.

### **4. What is Agile Testing?**

Testing practice for projects using agile methodologies, treating development as the customer of testing and emphasizing a test-first design paradigm. See also Test Driven Development.

### **5. What is Application Binary Interface (ABI)?**

A specification defining requirements for portability of applications in binary forms across different system platforms and environments.

### **6. What is Application Programming Interface (API)?**

A formalized set of software calls and routines that can be referenced by an application program in order to access supporting system or network services.

### **7. What is Automated Software Quality (ASQ)?**

The use of software tools, such as automated testing tools, to improve software quality.

### **8. What is Automated Testing?**

Testing employing software tools which execute tests without manual intervention. Can be applied in GUI, performance, API, etc. testing. The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

### **9. What is Backus-Naur Form?**



A metalanguage used to formally describe the syntax of a language.

**10. What is Basic Block?**

A sequence of one or more consecutive, executable statements containing no branches.

**11. What is Basis Path Testing?**

A white box test case design technique that uses the algorithmic flow of the program to design tests.

**12. What is Basis Set?**

The set of tests derived using basis path testing.

**13. What is Baseline?**

The point at which some deliverable produced during the software engineering process is put under formal change control.

**14. What you will do during the first day of job?**

What would you like to do five years from now?

**15. What is Beta Testing?**

Testing of a rerelease of a software product conducted by customers.

**16. What is Binary Portability Testing?**

Testing an executable application for portability across system platforms and environments, usually for conformation to an ABI specification.

**17. What is Black Box Testing?**

Testing based on an analysis of the specification of a piece of software without reference to its internal workings. The goal is to test how well the component conforms to the published requirements for the component.

**18. What is Bottom Up Testing?**

An approach to integration testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

**19. What is Boundary Testing?**

Test which focus on the boundary or limit conditions of the software being tested. (Some of these tests are stress tests).

**20. What is Bug?**

A fault in a program, which causes the program to perform in an unintended or unanticipated manner.

**20. What is Defect?**

If software misses some feature or function from what is there in requirement it is called as defect.

**21. What is Boundary Value Analysis?**

BVA is similar to Equivalence Partitioning but focuses on "corner cases" or values that are usually out of range as defined by the specification. This means that if a function expects all values in range of negative 100 to positive 1000, test inputs would include negative 101 and positive 1001.

## **22. What is Branch Testing?**

Testing in which all branches in the program source code are tested at least once.

## **23. What is Breadth Testing?**

A test suite that exercises the full functionality of a product but does not test features in detail.

## **24. What is CAST?**

Computer Aided Software Testing.

## **25. What is Capture/Replay Tool?**

A test tool that records test input as it is sent to the software under test. The input cases stored can then be used to reproduce the test at a later time. Most commonly applied to GUI test tools.

## **26. What is CMM?**

The Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes.

## **27. What is Cause Effect Graph?**

A graphical representation of inputs and the associated outputs effects which can be used to design test cases.

## **28. What is Code Complete?**

Phase of development where functionality is implemented in entirety; bug fixes are all that are left. All functions found in the Functional Specifications have been implemented.

## **29. What is Code Coverage?**

An analysis method that determines which parts of the software have been executed (covered) by the test case suite and which parts have not been executed and therefore may require additional attention.

## **30. What is Code Inspection?**

A formal testing technique where the programmer reviews source code with a group who ask questions analyzing the program logic, analyzing the code with respect to a checklist of historically common programming errors, and analyzing its compliance with coding standards.

## **31. What is Code Walkthrough?**

A formal testing technique where source code is traced by a group with a small set of test cases, while the state of program variables is manually monitored, to analyze the programmer's logic and assumptions.

## **32. What is Coding?**

The generation of source code.

**33. What is Compatibility Testing?**

Testing whether software is compatible with other elements of a system with which it should operate, e.g. browsers, Operating Systems, or hardware.

**34. What is Component?**

A minimal software item for which a separate specification is available.

**35. What is Component Testing?**

Testing of individual software components (Unit Testing).

**36. What is Concurrency Testing?**

Multi-user testing geared towards determining the effects of accessing the same application code, module or database records. Identifies and measures the level of locking, deadlocking and use of single-threaded code and locking semaphores.

**37. What is Conformance Testing?**

The process of testing that an implementation conforms to the specification on which it is based. Usually applied to testing conformance to a formal standard.

**38. What is Context Driven Testing?**

The context-driven school of software testing is flavor of Agile Testing that advocates continuous and creative evaluation of testing opportunities in light of the potential information revealed and the value of that information to the organization right now.

**39. What is Conversion Testing?**

Testing of programs or procedures used to convert data from existing systems for use in replacement systems.

**40. What is Cyclomatic Complexity?**

A measure of the logical complexity of an algorithm, used in white-box testing.

**41. What is Data Dictionary?**

A database that contains definitions of all data items defined during analysis.

**42. What is Data Flow Diagram?**

A modeling notation that represents a functional decomposition of a system.

**43. What is Data Driven Testing?**

Testing in which the action of a test case is parameterized by externally defined data values, maintained as a file or spreadsheet. A common technique in Automated Testing.

**44. What is Debugging?**

The process of finding and removing the causes of software failures.

**45. What is Defect?**

Nonconformance to requirements or functional / program specification

**46. What is Dependency Testing?**

Examines an application's requirements for pre-existing software, initial states and configuration in order to maintain proper functionality.

**47. What is Depth Testing?**

A test that exercises a feature of a product in full detail.

**48. What is Dynamic Testing?**

Testing software through executing it. See also Static Testing.

**49. What is Emulator?**

A device, computer program, or system that accepts the same inputs and produces the same outputs as a given system.

**50. What is Endurance Testing?**

Checks for memory leaks or other problems that may occur with prolonged execution

**51. What is End-to-End testing?**

Testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**52. What is Equivalence Class?**

A portion of a component's input or output domains for which the component's behaviour is assumed to be the same from the component's specification.

**53. What is Equivalence Partitioning?**

A test case design technique for a component in which test cases are designed to execute representatives from equivalence classes.

**54. What is Exhaustive Testing?**

Testing which covers all combinations of input values and preconditions for an element of the software under test.

**55. What is Functional Decomposition?**

A technique used during planning, analysis and design; creates a functional hierarchy for the software.

**54. What is Functional Specification?**

A document that describes in detail the characteristics of the product with regard to its intended features.

**55. What is Functional Testing?**

Testing the features and operational behavior of a product to ensure they correspond to its specifications. Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. or Black Box Testing.

**56. What is Glass Box Testing?**

A synonym for White Box Testing.

**57. What is Gorilla Testing?**

Testing one particular module, functionality heavily.

**58. What is Gray Box Testing?**

A combination of Black Box and White Box testing methodologies? testing a piece of software against its specification but using some knowledge of its internal workings.

**59. What is High Order Tests?**

Black-box tests conducted once the software has been integrated.

**60. What is Independent Test Group (ITG)?**

A group of people whose primary responsibility is software testing,

**61. What is Inspection?**

A group review quality improvement process for written material. It consists of two aspects; product (document itself) improvement and process improvement (of both document production and inspection).

**62. What is Integration Testing?**

Testing of combined parts of an application to determine if they function together correctly. Usually performed after unit and functional testing. This type of testing is especially relevant to client/server and distributed systems.

**63. What is Installation Testing?**

Confirms that the application under test recovers from expected or unexpected events without loss of data or functionality. Events can include shortage of disk space, unexpected loss of communication, or power out conditions.

**64. What is Load Testing?**

See Performance Testing.

**65. What is Localization Testing?**

This term refers to making software specifically designed for a specific locality.

**66. What is Loop Testing?**

A white box testing technique that exercises program loops.

**67. What is Metric?**

A standard of measurement. Software metrics are the statistics describing the structure or content of a program. A metric should be a real objective measurement of something such as number of bugs per lines of code.

**68. What is Monkey Testing?**

Testing a system or an Application on the fly, i.e just few tests here and there to ensure the system or an application does not crash out.

**69. What is Negative Testing?**

Testing aimed at showing software does not work. Also known as "test to fail". See also Positive Testing.

**70. What is Path Testing?**

Testing in which all paths in the program source code are tested at least once.

**71. What is Performance Testing?**

Testing conducted to evaluate the compliance of a system or component with specified performance requirements. Often this is performed using an automated test tool to simulate large number of users. Also known as "Load Testing".

**72. What is Positive Testing?**

Testing aimed at showing software works. Also known as "test to pass". See also Negative Testing.

**73. What is Quality Assurance?**

All those planned or systematic actions necessary to provide adequate confidence that a product or service is of the type and quality needed and expected by the customer.

**74. What is Quality Audit?**

A systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives.

**75. What is Quality Circle?**

A group of individuals with related interests that meet at regular intervals to consider problems or other matters related to the quality of outputs of a process and to the correction of problems or to the improvement of quality.

**76. What is Quality Control?**

The operational techniques and the activities used to fulfill and verify requirements of quality.

**77. What is Quality Management?**

That aspect of the overall management function that determines and implements the quality policy.

**78. What is Quality Policy?**

The overall intentions and direction of an organization as regards quality as formally expressed by top management.

**79. What is Quality System?**

The organizational structure, responsibilities, procedures, processes, and resources for implementing quality management.

**80. What is Race Condition?**

A cause of concurrency problems. Multiple accesses to a shared resource, at least one of which is a write, with no mechanism used by either to moderate simultaneous access.

**81. What is Ramp Testing?**

Continuously raising an input signal until the system breaks down.

**82. What is Recovery Testing?**

Confirms that the program recovers from expected or unexpected events without loss of data or functionality. Events can include shortage of disk space, unexpected loss of communication, or power out conditions

**83. What is Regression Testing?**

Retesting a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made.

**84. What is Release Candidate?**

A pre-release version, which contains the desired functionality of the final version, but which needs to be tested for bugs (which ideally should be removed before the final version is released).

**85. What is Sanity Testing?**

Brief test of major functional elements of a piece of software to determine if its basically operational. See also Smoke Testing.

**86. What is Scalability Testing?**

Performance testing focused on ensuring the application under test gracefully handles increases in work load.

**87. What is Security Testing?**

Testing which confirms that the program can restrict access to authorized personnel and that the authorized personnel can access the functions available to their security level.

**88. What is Smoke Testing?**

A quick-and-dirty test that the major functions of a piece of software work. Originated in the hardware testing practice of turning on a new piece of hardware for the first time and considering it a success if it does not catch on fire.

**89. What is Soak Testing?**

Running a system at high load for a prolonged period of time. For example, running several times more transactions in an entire day (or night) than would be expected in a busy day, to identify and performance problems that appear after a large number of transactions have been executed.

**90. What is Software Requirements Specification?**

A deliverable that describes all data, functional and behavioral requirements, all constraints, and all validation requirements for software/

**91. What is Software Testing?**

A set of activities conducted with the intent of finding errors in software.

**92. What is Static Analysis?**

Analysis of a program carried out without executing the program.

**93. What is Static Analyzer?**

A tool that carries out static analysis.

**94. What is Static Testing?**

Analysis of a program carried out without executing the program.

**95. What is Storage Testing?**

Testing that verifies the program under test stores data files in the correct directories and that it reserves sufficient space to prevent unexpected termination resulting from lack of space. This is external storage as opposed to internal storage.

**96. What is Stress Testing?**

Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. Often this is performance testing using a very high level of simulated load.

**97. What is Structural Testing?**

Testing based on an analysis of internal workings and structure of a piece of software. See also White Box Testing.

**98. What is System Testing?**

Testing that attempts to discover defects that are properties of the entire system rather than of its individual components.

**99. What is Testability?**

The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

**100. What is Testing?**

The process of exercising software to verify that it satisfies specified requirements and to detect errors. The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs), and to evaluate the features of the software item (Ref. IEEE Std 829). The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component. What is Test Automation? It is the same as Automated Testing.

**101. What is Test Bed?**

An execution environment configured for testing. May consist of specific hardware, OS, network topology, configuration of the product under test, other application or system software, etc. The Test Plan for a project should enumerate the test beds(s) to be used.

**102. What is Test Case?**

Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc. A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test Driven Development? Testing methodology associated with Agile Programming in which every chunk of code is covered by unit tests, which must all pass all the time, in an effort to eliminate unit-level and regression bugs during development. Practitioners of TDD write a lot of tests, i.e. an equal number of lines of test code to the size of the production code.

**103. What is Test Driver?**

A program or test tool used to execute tests. Also known as a Test Harness.

**104. What is Test Environment?**



The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers.

#### **105. What is Test First Design?**

Test-first design is one of the mandatory practices of Extreme Programming (XP). It requires that programmers do not write any production code until they have first written a unit test.

#### **106. What is Test Harness?**

A program or test tool used to execute a tests. Also known as a Test Driver.

#### **107. What is Test Plan?**

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

#### **108. What is Test Procedure?**

A document providing detailed instructions for the execution of one or more test cases.

#### **109. What is Test Script?**

Commonly used to refer to the instructions for a particular test that will be carried out by an automated test tool.

#### **110. What is Test Specification?**

A document specifying the test approach for a software feature or combination of features and the inputs, predicted results and execution conditions for the associated tests.

#### **111. What is Test Suite?**

A collection of tests used to validate the behavior of a product. The scope of a Test Suite varies from organization to organization. There may be several Test Suites for a particular product for example. In most cases however a Test Suite is a high level concept, grouping together hundreds or thousands of tests related by what they are intended to test.

#### **112. What is Test Tools?**

Computer programs used in the testing of a system, a component of the system, or its documentation.

#### **113. What is Thread Testing?**

A variation of top-down testing where the progressive integration of components follows the implementation of subsets of the requirements, as opposed to the integration of components by successively lower levels.

#### **114. What is Top Down Testing?**

An approach to integration testing where the component at the top of the component hierarchy is tested first, with lower level components being simulated by stubs. Tested components are then used to test lower level components. The process is repeated until the lowest level components have been tested.

#### **115. What is Total Quality Management?**

A company commitment to develop a process that achieves high quality product and customer satisfaction.

#### **116. What is Traceability Matrix?**

A document showing the relationship between Test Requirements and Test Cases.

#### **117. What is Usability Testing?**

Testing the ease with which users can learn and use a product.

#### **118. What is Use Case?**

The specification of tests that are conducted from the end-user perspective. Use cases tend to focus on operating software as an end-user would conduct their day-to-day activities.

#### **119. What is Unit Testing?**

Testing of individual software components.

#### **120. how do the companies expect the defect reporting to be communicated by the tester to the development team. Can the excel sheet template be used for defect reporting. If so what are the common fields that are to be included ? who assigns the priority and severity of the defect**

To report bugs in excel:

Sno. Module Screen/ Section Issue detail Severity

Prioriety Issuestatus

this is how to report bugs in excel sheet and also set filters on the Columns attributes.

But most of the companies use the share point process of reporting bugs In this when the project came for testing a module wise detail of project is inserted to the defect managment system they are using. It contains following field

1. Date
2. Issue brief
3. Issue discription(used for developer to regenrate the issue)
4. Issue satus( active, resolved, onhold, suspend and not able to regenrate)
5. Assign to (Names of members allocated to project)
6. Prioriety(High, medium and low)
7. Severity (Major, medium and low)

#### **121. How do you plan test automation?**

1. Prepare the automation Test plan
2. Identify the scenario
3. Record the scenario
4. Enhance the scripts by inserting check points and Conditional Loops
5. Incorporated Error Hnadler
6. Debug the script
7. Fix the issue
8. Rerun the script and report the result

#### **122. Does automation replace manual testing?**

There can be some functionality which cannot be tested in an automated tool so we may have to do it manually. therefore manual testing can never be replaced. (We can write the scripts for negative testing also but it is hectic task).When we talk about real environment we do negative testing manually.

#### **123. How will you choose a tool for test automation?**

choosing of a tool deends on many things ...

1. Application to be tested
2. Test environment
3. Scope and limitation of the tool.
4. Feature of the tool.
5. Cost of the tool.
6. Whether the tool is compatible with your application which means tool should be able to interact with your appliaction
7. Ease of use

#### **124. How you will evaluate the tool for test automation?**

We need to concentrate on the features of the tools and how this could be beneficial for our project. The additional new features and the enhancements of the features will also help.

**125. How you will describe testing activities?**

Testing activities start from the elaboration phase. The various testing activities are preparing the test plan, Preparing test cases, Execute the test case, Log the bug, validate the bug & take appropriate action for the bug, Automate the test cases.

**126. What testing activities you may want to automate?**

Automate all the high priority test cases which needs to be executed as a part of regression testing for each build cycle.

**127. Describe common problems of test automation.**

The common problems are:

1. Maintenance of the old script when there is a feature change or enhancement
2. The change in technology of the application will affect the old scripts

**128. What types of scripting techniques for test automation do you know?**

5 types of scripting techniques:

Linear  
Structured  
Shared  
Data Driven  
Key Driven

**129. What is memory leaks and buffer overflows ?**

Memory leaks means incomplete deallocation – are bugs that happen very often. Buffer overflow means data sent as input to the server that overflows the boundaries of the input area, thus causing the server to misbehave. Buffer overflows can be used.

**130. What are the major differences between stress testing,load testing,Volume testing?**

Stress testing means increasing the load ,and cheking the performance at each level. Load testing means at a time giving more load by the expectation and cheking the performance at that leval. Volume testing means first we have to apply initial.