- Drawbacks:
  1. Costly tool
  2. All the areas can't be automated.

- **Types of Automated Tool:**

  Automated tools can be broadly divided into 3 types:
  1. Functional Tools: QTP,Selenium,Win Runner,…
  2. Management Tools: QC,Test Director,…
  3. Performance Tools: Load Runner,Silk Performer,…

- **QTP Introduction:**

1. Types of the tool: Functional Tool.
2. Company: Taken by Mercury corporation and handled by HP.
3. Versions: 5.5,6.5,7.6,8.0,8.2,9.0,9.2,9.5,10.0,11.0 (latest 11.5 beta)
4. Scripting languages: VB Script/Java Script.

- **Add-in Manager:** It is provided by the QTP,if you want to enter QTP, we need to select anyone of the Add-in based on application.By default, its having three types of Add-in

  ActiveX and visual basic for windows based applications.

  Web for web based applications.

- **Object Repository:** It's a storage place. The properties and values will be stored under object repository. It can be divided into two types.
  1. Local Repository.
  2. Shared Repository.

1. Local Repository: By default, all the properties and values will be stored under local repositories.

   NOTE1: Local repository extension .mtr (mercury test repository)

   NOTE2: Shared repository extension .tsr (test share repository).

   NOTE3: We can modify local repository, but we cannot modify shared repository.

   NOTE4: Local repositories we cannot associate for multiple test and for shared repositories we can associate.

- **Navigation for creating the shared repositories:**

1. Go to the object repository, activate the file menu. Select the option, export local objects, specify the file name with the extension .tsr.
2. Go to the object repositories, tools menu, select the option associate repositories, the associate repositories which cannot be opened, click on add repository button and select the required .tsr file and select the available actions, click on associate button, click on OK.

   NOTE1: Above 2nd paragraph is for, only if any local repositories files are deleted)

   Shortcut key: object repository (ctrl + R)

2. Identifying the areas to be automated.

3. Analyzing the both positive and negative flow of the application.

4. Based on the above analysis he will prepare the automation test plan document.

5. He will prepare the tool ready for the operations with all the pre configurational settings.

2. Generating the basic Script: In this phase the automation test engineer will generate the basic test for both positive and negative flow of the application.

3.Enhancing the Test:

1. Data Driven Test with parameterization:

DDT: It is a concept provided in Automation inorder to implement retesting.

Steps to be followed to perform DDT.

1.Collects the data into the data table.

2.Generate the basic test.

3.Parameterize the test.

4.Execute the test.

5.Analyse the result.

2. Parameterization: It is a process of replacing the constants with the variables or parameters inorder to increase the scope of the test.

It can be done in 6 ways:

1.Through Data Driven Wizard.

2.Through Keyword view.

3.Manually.

4.Flat Files, Databases and Excel files(Adv. QTP).

5.Passing the input and output parameters.

6.Passing the input values during execution.

7.Declaring User Defined Variavles.

1.Through Data Driven Wizard:

Navigation:

2.Select the checkpoint type as Database Checkpoint.

3.Select the option specify SQL statement manually.

4.Click on Next button.

5.The database query wizard will be opened.

6.Click on create button to create the new connection string.

7.The database wizard will be opened.

8.Click on new button.

9.Select the database type.

10.Click on next.

11.Click on browse button and specify the data source name.

12.Save ODBC Microsoft Access Setup Wizard will be opened.

13.Select the Database, click on ok.

14.Now select the data source name.

15.Now specify the SQL statement.

16.Select SQL statement Select *from tablename

17.Click on OK.

6. **XML Checkpoint:** XML checkpoint is used to checking the content of the XML.
7. **Accessibility Checkpoint:** It is a web based checkpoint, it is used for checking whether the application is developed as per “www” standards.

Output Values: Capturing something during execution, we can findout result in the runtime data table, QTP has provided 4 types of output values

1. **Standard Output:** It is used for capturing the specific object properties and values.
2. **Database Output:** It is used to capturing the contents from the datatable.
3. **XML Output:** It is used to capturing the content from the XML file.
4. **Text Output:** It is used for checking the text present on a specified object.
5. **Text Area Output:** It is used for checking the text present in a specified area.

Example for Standard Output Value:

VbWindow(“Form”1).VbButton(“ADD”).Output CheckPoint(“ADD”)

Example for Database Output:

4. Import Sheet: It is used for importing a specified sheet of the data from the excel file to a specified sheet of data in the run time data table.

Syntax: Datatable.ImportSheet “Path of the excel file, SourceSheet id, DestinationSheet id”

5. Export: It is used for exporting the data present in the run time data table to a specified location.

Syntax: Datatable.Export “Path of the Excel file”

6. Export Sheet: It is used for exporting a specified sheet of the data from the run time data table to a specified location.

Syntax: Datatable.Export “Path of the Excel File”, Sheet id to be exported.

7. SetCurrentRow: It is used for making the QTP focus on a specified Row.

Syntax: Datatable.SetCurrentRow(RowNumber)

8. SetNextRow: It is used for making the QTP focus on the Next Row of the currently focused row.

Syntax: Datatable.SetNextRow

9. SetPrevRow: It is used for making the QTP focus on the previous row of the currently focused row.

Syntax: Datatable.SetPrevRow

10. Value Method: Value method is used for getting a value from a specified sheet, specified column and currently focused row.

11. GetSheet: It is used for making the QTP focus on a specified sheet.

Syntax: Datatable.GetSheet(SheetID)

12. GetRowCount: It is used for getting the rowcount.

NOTE:

By default it will return the global sheet row count.

var = Datatable.GetRowCount

Example:

```
Datatable.AddSheet “roopa”

Datatable.ImportSheet “d:\td.xls”,1,3

n= Datatable.GetSheet(3).GetRowCount(i)

VbWindow(“Form1”).VbEdit(“Text1”).Set Datatable.Value(“V1”,3)

VbWindow(“Form1”).VbEdit(“Text2”).Set Datatable.Value(“V2”,3)

VbWindow(“Form1”).VbButton(“ADD”).Click

Actual = VbWindow(“Form1”).VbEdit(“Text3”).GetROProperty(“Text”)

Expval = Datatable.Value(“ev”,3)

If Cint(Expval) = Cint(Actual) Then
```

Dim Fso,F1

Set Fso = CreateObject(“Scripting.FileSystemObject”)

Set F1= Fso.OpenTextFile(“c:\d and s|s1|desktop\1234.txt”,1)

Msgbox ReadAll

2.Write a program to write data into a text.

NOTE:

CreateTextFile method: This method is used to creating a text file to a specific location.

Syntax: .CreateTextFile(“txt file location”)

WriteLine method: This method is used to write the data line by line.

Write method: This method is used to write the data character wise.

WriteBlankLines method: This method is used to write the blank lines.

Example:

Dim Fso,F1

Set Fso = CreateObject(“Scripting.FileSystemObject”)

Set F1 = Fso.CreateTextFile(“c:\d and s\s1\Desktop\result.txt”,2)

F1.Write(“Good Morning”)

F1.WriteBlankLines(10)

F1.WriteLine(“How are you”)

F1.WriteLine(“Welcome to India”)

a)Do the parameterization for flight reservation window through scripting (If the customer is provided the data in the flat file)

Step1: Prepare the test data in the following name.

D_flight@fly_from@fly_to@c-name

121222@London@Denver@roopa

Virtual Object Configuration: It is not a real object, but it will work like real object. Whenever the QTP not able to identify the object, then we go for virtual object configuration.

Virtual Object Manager: It is used for storing the virtual object information and also provides a provision to delete unnecessary virtual objects.

Navigation to create virtual objects:

- Activate the menu item.
- Tools, virtual objects.
- New virtual object.
- The virtual object manager wizard will be opened.
- Click on next.
- Select the object type.
- Click on next.
- Select the area with the help of mark object.
- Click on next.
- Select the parent hierarchy.
- Click on next.
- Specify the name of the button.
- Click on finish.

Regular Expressions: Whenever the object value is changing dynamically, then we will go for regular expressions.

Example:

Window("Flight Reservation:").Dialog("Fax Order No.*").ActiveX("MaskEdBox").Type "1111111111"

Window("Flight Reservation:").Dialog("Fax Order No.*")

WinButton("send").Click

Window("Flight Reservation:").Activate

Tips for generating the basic script for web based applications.

Object/Browser	**Script**
Browser	Browser(“Browser Name”)
Page	Page(“Page Name”)
EditBox	WebEdit(“EditBoxName”)
RadioButton	WebRadioButton(“RadioButton Name”)
CheckBox	WebCheckBox(“CheckBox Name”)
ListBox	WebList(“ListBox Name”)

Example:

Capture the link name from the google web page and export into an text file.

NOTE:

Child objects: This method is used to capture the classname for specific object.

micclass method: This method is used to capture the name of the object.

ObjectCount: To count the number of specific objects.

```
Dim Fso,Myfile,Odesc,Lists,Nlinks
Set Fso = createobject(“Scripting.FileSystemObject”)
Set Myfile = Fso.CreateTextFile(“d:\linknames.txt”,2)
Myfile.WriteLine “linknames”
Myfile.WriteLine “*******”
Set Odesc = description.Create()
oDesc(“micclass”).Value = “link”
Set lists = Browser(“Google”).Page(“Google”).ChildObjects(Odesc)
Nlinks = lists.Count()
msgbox Nlinks
```