

Basics of Automation and Overview of QTP

By,

Anver Sathic

Abdul Subhan

AGENDA

- Manual Testing
 - Con's
- Automation Testing
 - Pro's and Con's
- Automation Testing tools
- Automation Planning
- QTP at a Glance & Add-In Manager
- QTP Testing Workflow
- Different Test pane views in QTP
- QTP User Interface
- QTP Testing Process Explained



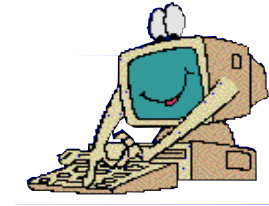
Why automation?



No Testing



Manual Testing



Automated Testing

- × Time consuming
- × Not always reliable
- × Hard to Repeat
- × Costly (More resources)
- × Inconsistent

- ✓ Consistent
- ✓ Reduction in test cycle time
- ✓ Reliable
- ✓ Reusability of test wares
- ✓ Good ROI

Do Automation !



- Tests that need to be run for every build of the application (sanity check, regression test)
- Regression Testing cycles are long and iterative
- Application Under Test is planned to have multiple releases/builds
- Tests that use multiple data values for the same actions (data driven tests)
- Long running application where in small enhancements/Bug fixes keeps happening
- Test Repeatability is required



More repetitive execution?

Don't Automate !

- Usability testing.
 - "How easy is the application to use?"
- One-time testing.
- "ASAP" testing.
 - "We need to test NOW!"
- Ad hoc/random testing.
 - Based on intuition and knowledge Application
- Test without predictable results.



Improvisation required?

DIFFERENT AUTOMATION TOOLS

- **QaCenter : Compuware**
- **Silk Test : Segue Software Inc**
- **Rational Robot : Rational Software**
- **QTP : HP**



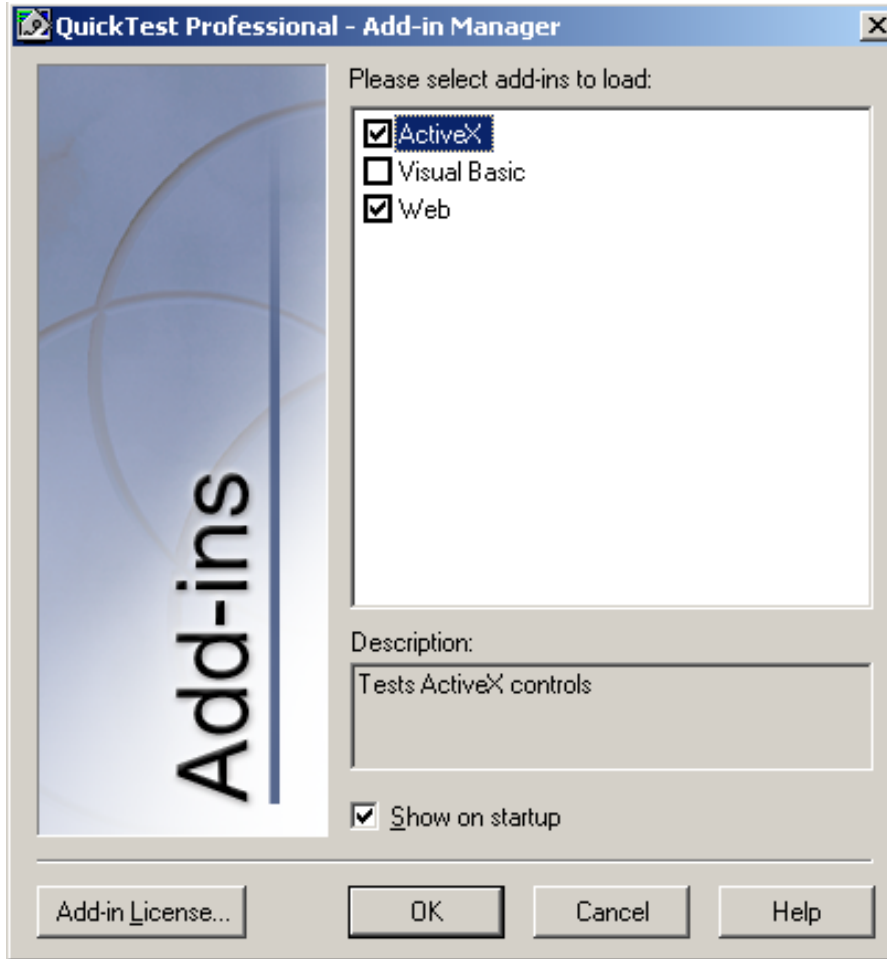
What is QTP?

- HP Quick Test Pro is a functional automation and regression testing tool.

Technologies Supported

- Quick Test Professional enables you to test standard Windows applications, Web objects, ActiveX controls, Visual Basic applications, and multimedia objects on Web pages.
- Default Support
 1. Standard Windows applications
 2. Web Objects/applications
 3. ActiveX Controls
 4. Visual Basic Applications

Add-in Manager

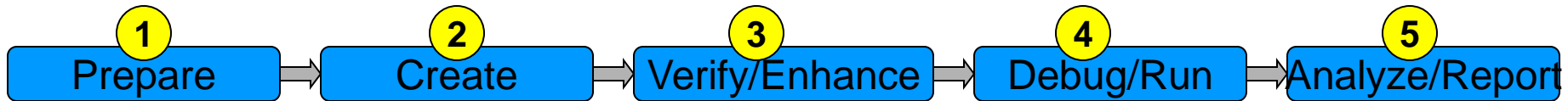


- Default Add-ins
 - ActiveX
 - Visual basic
 - Web

- Other Add-Ins
 - Siebel
 - Java
 - SAP
 - Oracle
 - .Net and many more

QuickTest Professional Workflow

Main phases of QTP testing process:



Setup for Test Automation	Create basic test	Enhance basic test	Debug/Run test	Analyze/Report
<ul style="list-style-type: none">•Document manual test steps•Check application/environment stability•Check that data is Valid	<ul style="list-style-type: none">•Record user actions/session in Application or website•Build Object Repository and use objects to add steps manually	<ul style="list-style-type: none">•Adding Logical and conditional statements•Insert Checkpoints•Parameterization	<ul style="list-style-type: none">•Check if test script operates smoothly & without any interruption, if any changes were made•Running test to check the behavior	<ul style="list-style-type: none">•Examine test results to pinpoint defects in application•Create defect reports if any failures are encountered during analysis

Different Panes in QTP

■ Test Pane

Test Pane contains two tabs to view the tests,

Keyword View- Actions are generated in tabular format similar to manual testcases. This is often the easiest code view for non-programmers.

Expert View- Actions are generated in VB Scripting code based representations. This is the most flexible code.

■ Active Screen

- Provides screenshots of all objects recorded and also supports object highlighting.

■ Data Table

- Shows the data file that can be used to configure the application

■ Debug Viewer

- Provides tool to debug code and view the value of variable/expression during code execution.

QTP - Main Window

The screenshot shows the main window of QuickTest Professional (QTP) with several components labeled:

- Title Bar:** Parameter - QuickTest Professional
- Menu Bar:** File Edit View Insert Test Step Debug Tools Help
- File Toolbar:** Contains icons for File, Edit, View, Insert, Test, Step, Debug, Tools, and Help.
- Debug Toolbar:** Contains icons for Run, Stop, and other debugging functions.
- Testing Toolbar:** Contains icons for Record, Run, Stop, and other testing functions.
- Keyword View:** A table showing the test flow with columns for Item, Operation, Value, Comment, and Documentation.
- Data Table:** A table with columns A, B, and C, and rows 1 through 8. Row 1 contains 'Sandra', row 2 'Anthony', and row 3 'Natasha'.
- Active Window:** A dialog box titled 'Login' with fields for 'Agent Name' (containing 'Laurence') and 'Password', and buttons for 'OK', 'Cancel', and 'Help'.
- Status Bar:** Shows 'Global / Action1' and 'Active Screen'.

Item	Operation	Value	Comment	Documentation
▼ Action1				
▼ Login				
Agent Name: Set		"Laurence"		Enter "Laurence" in the "Agent Name:" edit box.
Password: SetSecure		"407d88a186717fb56d77..."		Enter the encrypted string "407d88a186717fb56d77..."
OK	Click			Click the "OK" button.
Flight Reservation	Activate			Make the "Flight Reservation" window active.

	A	B	C
1	Sandra		
2	Anthony		
3	Natasha		
4			
5			
6			
7			
8			

QuickTest Professional Keyword View

ICON-BASED KEYWORD VIEW
Business analysts and new users can quickly generate automated tests with no programming code.

Item	Operation	Value	Documentation
▼ Action1			
▼ Login			
Agent Name:	Set	"jojo"	Enter "jojo" in the "Agent Name:" edit box.
Password:	SetSecure	"409a78ba0e283ab55c2a8eeae9cce2..."	Enter the encrypted string "409a78ba0e283ab55c..."

DATA TABLE
Different data can be used to run one test (instead of manually running tests repeatedly)

	A	B
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

ACTIVE SCREEN
Interact with objects captured during recording

Item	Operation	Value	Documentation
▼ Action1			
▼ Welcome: Mercury Tours			
▼ Welcome: Mercury ...			
userName	Set	"mercury"	Enter "mercury" in the "userName" edit box.
password	SetSecure	"4082986e39ea..."	Enter the encrypted string "4082986e39ea469e70dbf8c5..."
Sign-In	Click	2,2	Click the "Sign-In" image.
▼ Find a Flight: Mercury			
fromPort	Select	"New York"	Select the "New York" item in the "fromPort" list.
fromMonth	Select	"Dec"	Select the "Dec" item in the "fromMonth" list.
fromDay	Select	"29"	Select the "29" item in the "fromDay" list.
toPort	Select	"San Francisco"	Select the "San Francisco" item in the "toPort" list.
toMonth	Select	"Dec"	Select the "Dec" item in the "toMonth" list.
toDay	Select	"31"	Select the "31" item in the "toDay" list.
servClass	Select	"Business"	Select radio button "Business" in the "servClass" radio button group.
findFlights	Click	2,2	Click the "findFlights" image.

QuickTest Professional Expert View

```
Action1
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").WebEdit("password").SetSecure
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").Image("Sign-In").Click 2,2
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("fromPort").Select "New York"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("fromMonth").Select "Dec"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("fromDay").Select "29"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("toPort").Select "San Francisco"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("toMonth").Select "Dec"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebList("toDay").Select "31"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").WebRadioGroup("servClass").Select
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").Image("findFlights").Click 2,2
Browser("Welcome: Mercury Tours").Page("Select a Flight: Mercury").Image("reserveFlights").Click 2,2
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebEdit("passFirst0").Set "Nicole"
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebEdit("passLast0").Set "Jones"
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebEdit("creditnumber").Set "12345"
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebCheckBox("ticketLess").Set "ON"
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").Image("buyFlights").Click 2,2
Browser("Welcome: Mercury Tours").Page("Flight Confirmation: Mercury").Image("home").Click
```

QTP Testing Process in Brief

First Step – Preparing/Planning

Before starting to build a test, proper planning is to be done to Prepare the automated test environment.

Points to be considered:

Detailed description of the test with the exact steps to follow.

Environment/application stability.

Determine the initial and end conditions of the test.

Data to be Input.

Decide on how to organize the Object Repositories.

All items to be verified by the test.

Creating a Test

Second Step – Create Tests or Components

Create a test or component by,

- **Either recording a session on application or Website**

As navigating through the application or site, Quick Test graphically displays each step we perform as a row in the Keyboard View. The documentation column of the Keyboard View also displays a description of each step in easy-to-understand sentences. A step is something that causes or makes a change in the site or application, such as clicking a link or image, or submitting a data form.

- **Building Object Repository**

Build an object repository and use these objects to add steps manually in the Keyboard View or Expert View. It's possible to modify the test or components with special testing options and/or with programming statements. QuickTest has two types of object repositories for storing object information: ***shared object repositories*** and ***action object repositories***.

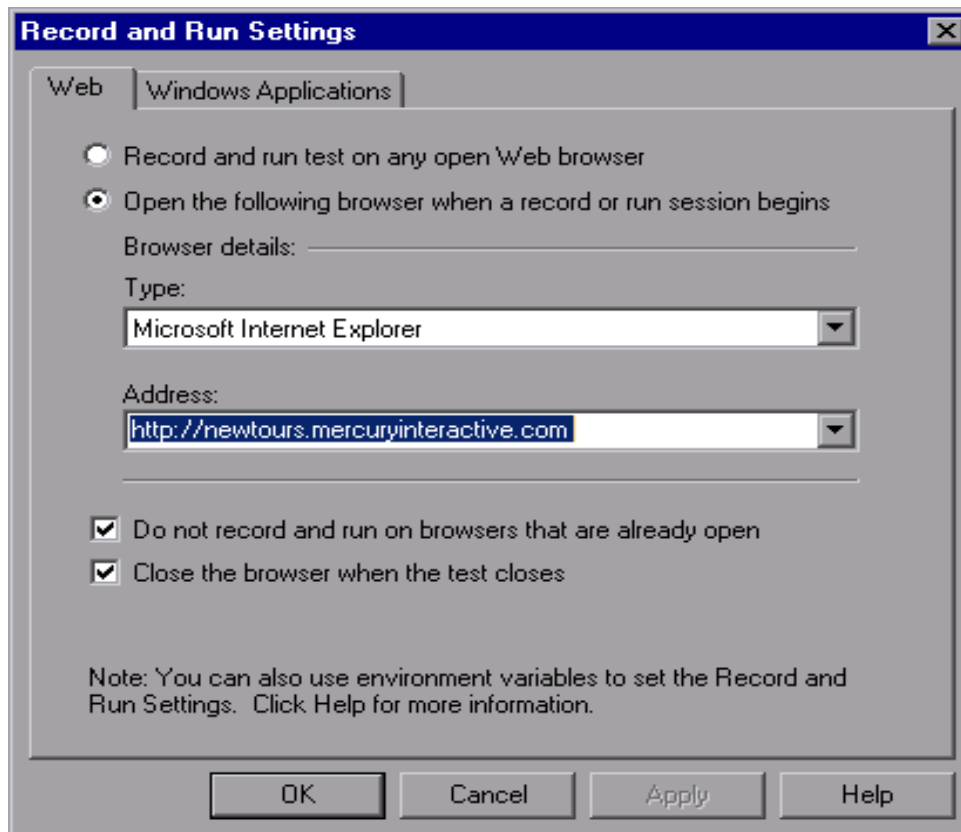
Recording a Test or Component

To start recording on,

Choose **Test (New Button** for a new test **or File > Open** to open an existing test).

Choose **Record** or click the **Record button**. 

Record and Run setting dialog box opens. Click **OK** to close it and begin recording.



Recording a Test or Component

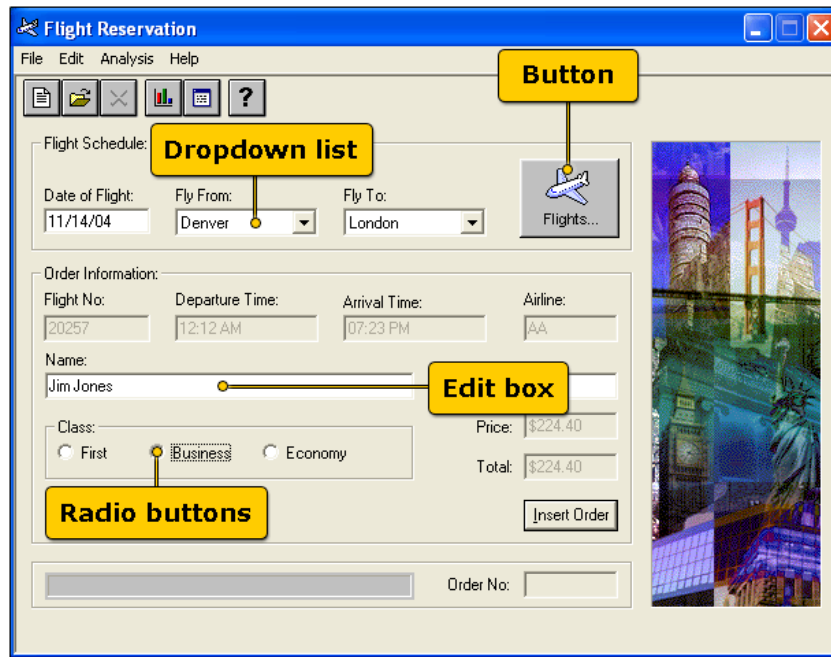
While recording, QuickTest creates a graphical representation of the steps performed on the application. These steps are displayed in the Keyword View Tab.

Item	Operation	Value	Documentation
▼ Action1			Call the Action1 action.
▼ Welcome: Mercury Tours			
▼ Welcome: Mercury Tours			
editBox userName	Set	"mercury"	Enter "mercury" in the "userName" edit box.
editBox password	SetSecure	"405d7bdbe..."	Enter the encrypted string "405d7bdbe1b2..."
Image Sign-In	Click	2,2	Click the "Sign-In" image.

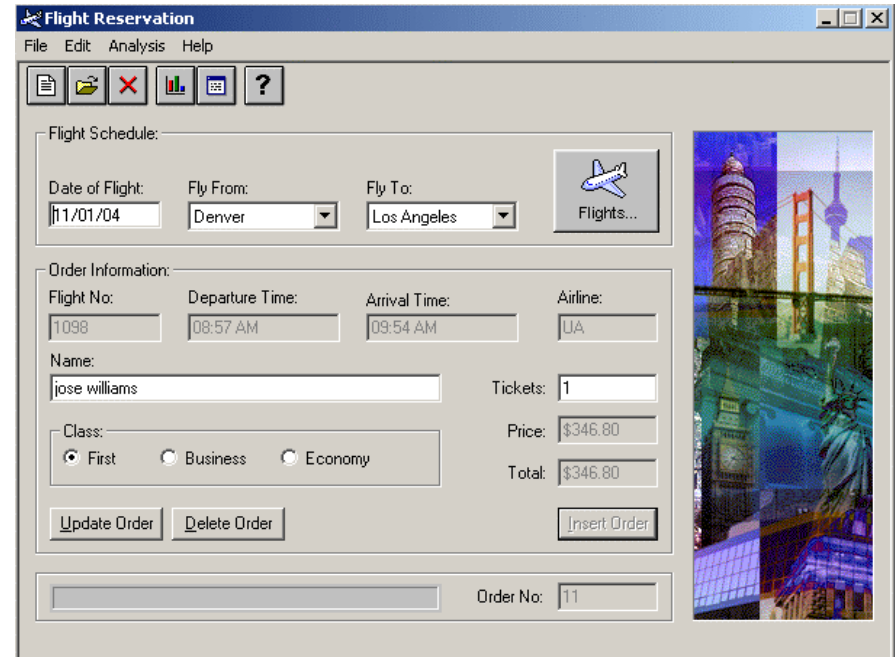
Object Repository

QuickTest stores the definitions for application objects in a file called the Object Repository. As you record your test, QuickTest will add an entry for each item you interact with. Each Object Repository entry will be identified by a logical name (determined automatically by QuickTest), and will contain a set of properties (type, name, etc) that uniquely identify each object.

Object Types



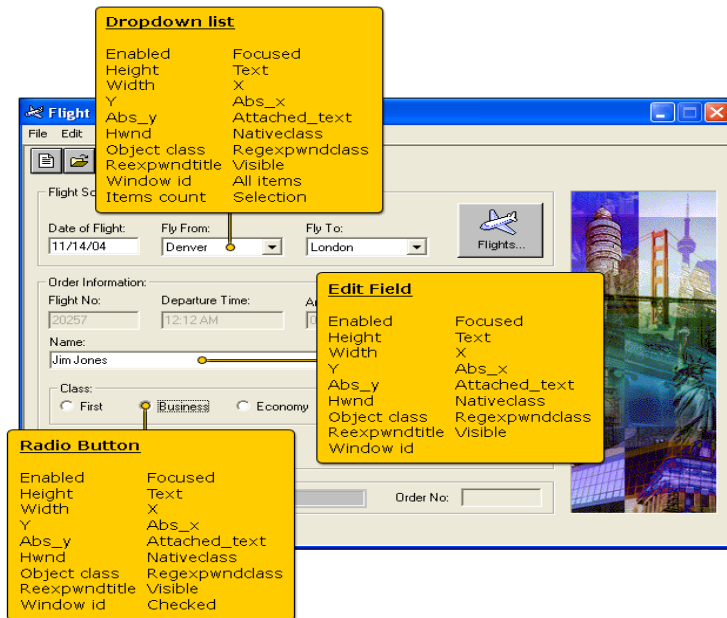
Object Properties



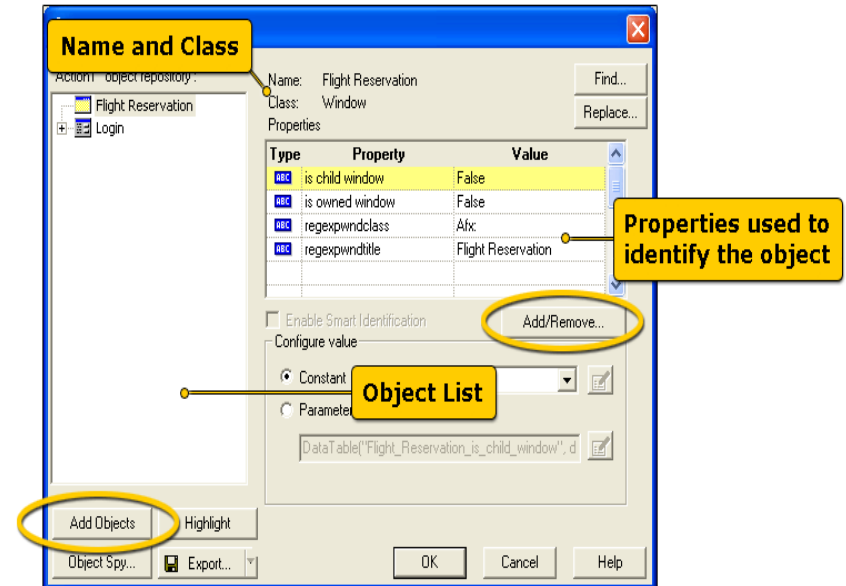
Object Identification

QTP Object Repository displays a tree of all objects in the current component or in the current action or entire test (depending on the object repository mode you selected). We can view or modify the test object description of any test object in the repository or to add new objects to the repository. Quicktest learns the default property values and determines in which test object class it fits. If it is not enough it adds assistive properties, one by one to the description until it has compiled the unique description. If no assistive properties are available, then it adds a special Ordinal identifier such as objects location on the page or in the source code.

Objects-Property/value Pairs



Stored Objects



Enhance Tests

Third Step – Enhancing Tests

Enhance the automated test for playback and testing

- **Inserting checkpoints into the test or component**

- A checkpoint is a verification point that compares a recent value for a specified property with the expected value for that property. This enables us to identify whether the Web site of application is functioning correctly.
- It lets us search for a specific value of a page, object or text string, to test for the proper functionality of the application.
- Checkpoints can be added to a test as we record or afterwards via the Active Screen.

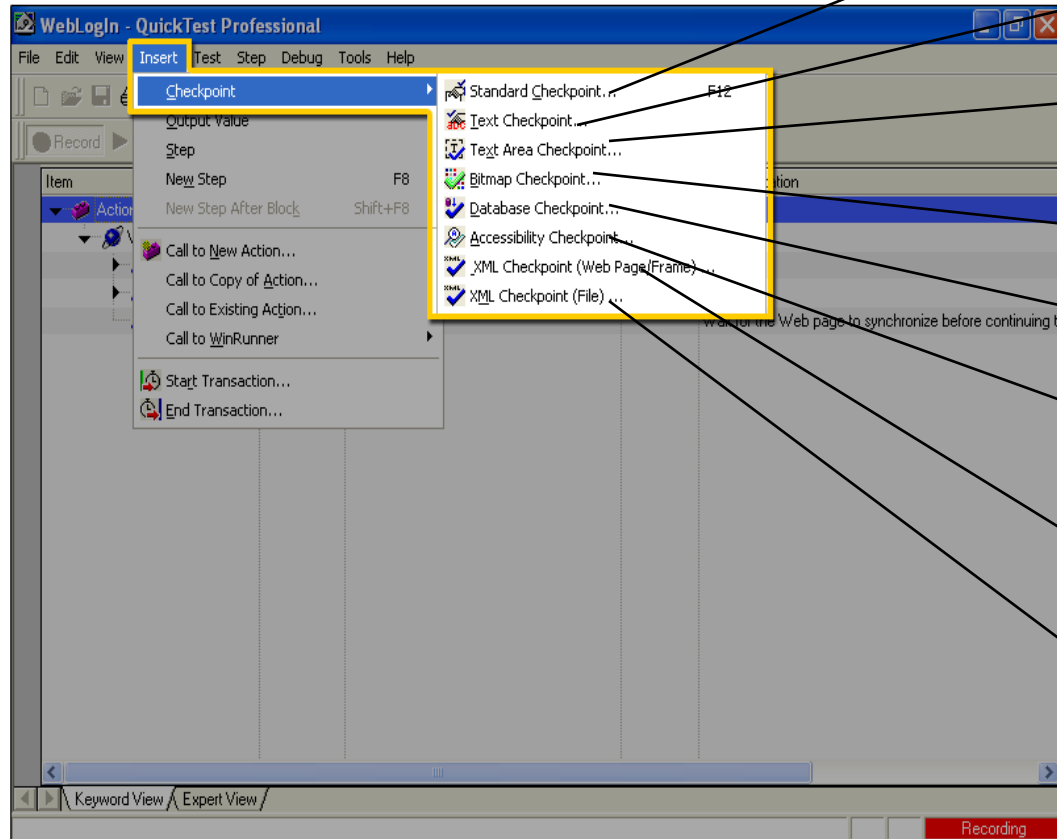
- **Inserting synchronization points**

- Problem due to mismatch between tool time and application time is sorted using synchronization.
- Used to slowdown the execution of the script.
- Waits until all objects are downloaded in a webpage.

Checkpoints and Types

A checkpoint is a specialized step in QuickTest that compares two values and reports the result.

Types of Checkpoints:



Checks the property value of an object in application or Web page.

Checks whether text string is displayed in the app. place in webpage.

Checks a text string is displayed within a defined area in a windows applcn, accord to specified criteria.

Checks an area of webpage or application as a bitmap.

Checks the contents of databases accessed by the website.

Identifies areas of Web site that may not conform to W3C Web content accessibility guidelines.

Checks the data content of XML documents in web pages and frames.

Checks the data content of XML documents in XML files or web pages and frames.

Synchronization Points

When you run tests, your application may not always respond with the same speed. For example, it might take a few seconds:

Why Synchronize?

- for a progress bar to reach 100%
- for a status message to appear
- for a button to become enabled
- for a window or pop-up message to open

It's possible to handle these anticipated timing problems by synchronizing the test to ensure that QuickTest waits until the application is ready before performing a certain step.

Methods:

- Sync
- Exist
- Wait Property
- Wait

Fourth Step – Debugging/Running Tests

- Running the script to debug it and ensure that it operates smoothly and without interruption, if any changes were made to the script.
- While running ,Quick Test connects to the application and perform each operation in a test or component.
- Possible to control the run session to identify and eliminate defects in the test or component, we can use Step Into, Step Over & Step Out commands to run a test or component step by step.
- Possible to set breakpoints to pause the test script at pre-determined points.
- Watch List, mainly used for better debugging to view the value of the variables/parameters during runtime, each time when it stops at a breakpoint in the debug viewer.

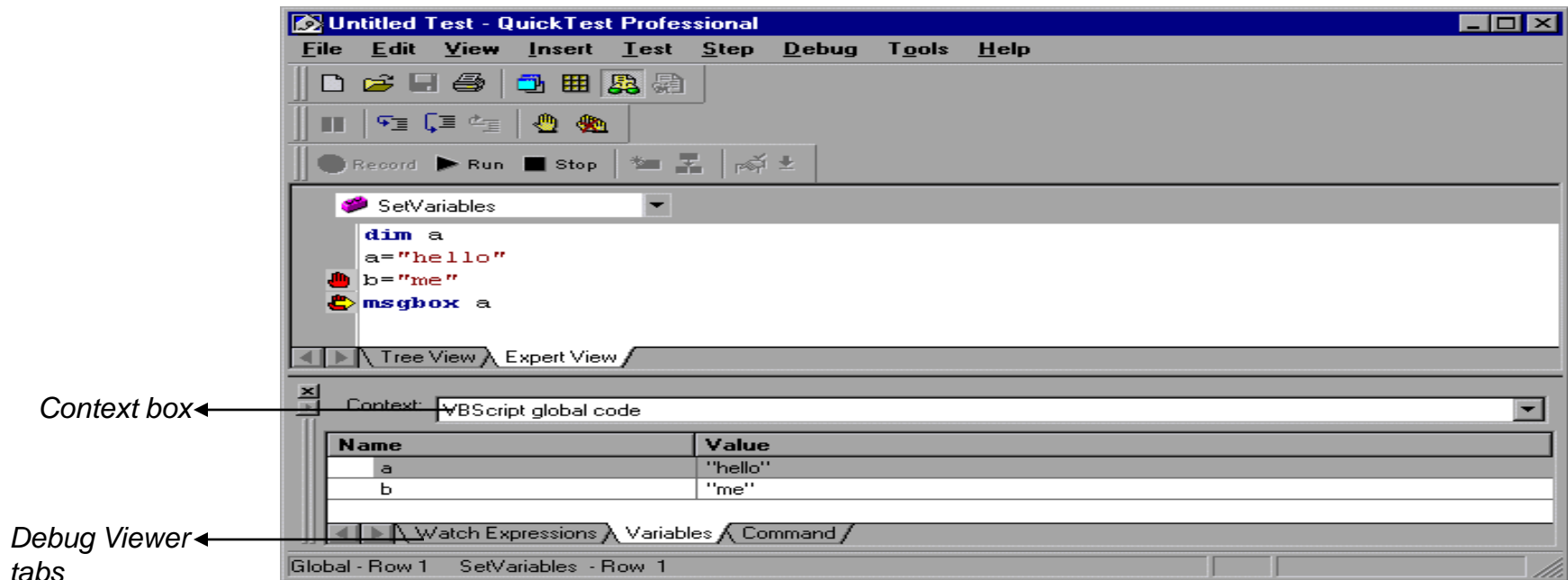
Using the Debug Viewer

When a test stops at a breakpoint, you use the Debug Viewer pane to view, set, or modify the current value of objects or variables in your test.

To open the Debug Viewer pane:

Run a test with one or more breakpoints.

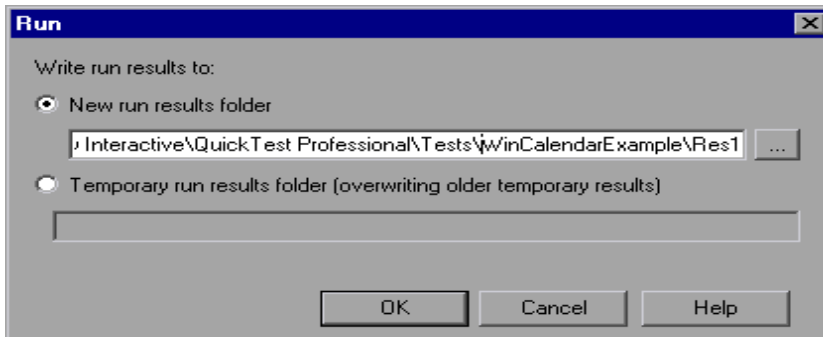
When the test pauses at a breakpoint, choose **View > Debug Viewer** or Click the Debug Viewer button. The Debug Viewer pane opens along the bottom of the QuickTest screen.



Run a Test

To run a test to check your application:

- If your test is not already open, choose **File > Open** or click the **Open** button to open the test.
- Click the **Run** button on the toolbar, or choose **Test > Run**. The Run dialog box opens, with **New run results folder** selected by default. This option displays the default path and a folder name in which the test run results are saved.



- To save the test run results in a different folder, type the path in the text box or click the browse button to locate the folder. If you are running a test from a TestDirector project, the **Run name**, **Test set**, and **Instance** options are displayed instead of the **New run results folder** box.
- To save the test run results in a temporary folder, click **Temporary run results folder**. This overwrites any results previously saved in this folder.

Analyze/Report

Fifth Step – Analyze/Report

After we run the test or component, we can view the results.

View the results in the Results window

After running the test or component, the results of the run can be viewed in the Test Results window. It's also possible to view a summary of the results as well as a detailed report.

Reporting Defects

If Quality Center is installed, it's possible to report the defects found out to a database. Also, possible to instruct Quick Test to automatically report each failed step in the test or component, or can report them manually from the test results window.

Analyzing Run Results

When Quick Test finishes running the test, the Test Results window opens by default.

The screenshot shows the 'MercTours1 [Res1] - Test Results' window. It features a title bar, a menu bar (File, View, Tools, Help), and a toolbar with various icons. On the left is a 'Test results tree' showing a hierarchy: 'Test MercTours1 Summary' (expanded), 'Run-Time Data', and three iterations: 'MercTours1 Iteration 1 (Passed)', 'MercTours1 Iteration 2 (Failed)', and 'MercTours1 Iteration 3 (Failed)'. The main area displays 'MercTours1 Results Summary' with the following details:

- Test:** MercTours1
- Results Name :** Res1
- Run started:** 7/3/02 - 11:37:25
- Run ended:** 7/3/02 - 11:38:25

Iteration #	Results
<u>1</u>	Passed
<u>2</u>	Failed
<u>3</u>	Failed

Status	Times
Passed	10
Failed	2
Warnings	0

At the bottom, a status bar shows 'For Help, press F1' and 'Ready'.

Labels with arrows pointing to the window components:

- Test results title bar
- Menu bar
- Test results tool bar
- Test results tree
- Test results details
- Status bar



Thank you