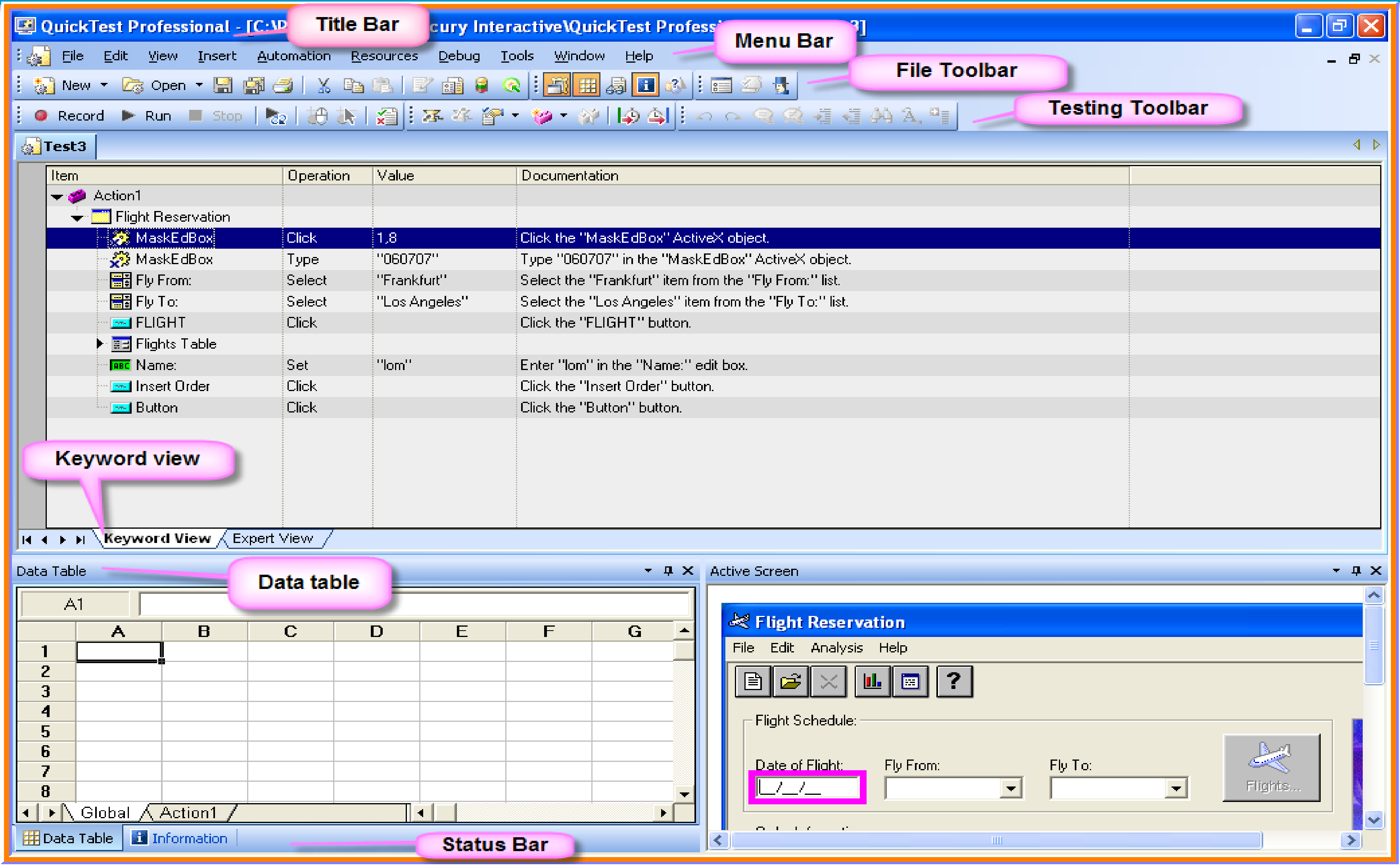
CONSULTING.TECHNOLOGY.OUTSOURCING

Basic QTP

People matter, results count.

User Interface of QTP



Item	Operation	Value	Documentation
Action1			
Flight Reservation			
MaskEdBox	Click	1,8	Click the "MaskEdBox" ActiveX object.
MaskEdBox	Type	"060707"	Type "060707" in the "MaskEdBox" ActiveX object.
Fly From:	Select	"Frankfurt"	Select the "Frankfurt" item from the "Fly From:" list.
Fly To:	Select	"Los Angeles"	Select the "Los Angeles" item from the "Fly To:" list.
FLIGHT	Click		Click the "FLIGHT" button.
Flights Table			
Name:	Set	"lom"	Enter "lom" in the "Name:" edit box.
Insert Order	Click		Click the "Insert Order" button.
Button	Click		Click the "Button" button.

Recording modes

Normal Recording – Is default mode of recording. It recognizes objects in the application regardless of their location on the screen. It records the objects and actions performed on them.

Analog Recording— Enables you to record the exact mouse and keyboard operations you perform in relation to either the screen or the application window. In this recording mode, QTP records and tracks every movement of the mouse as you drag the mouse around a screen or window.

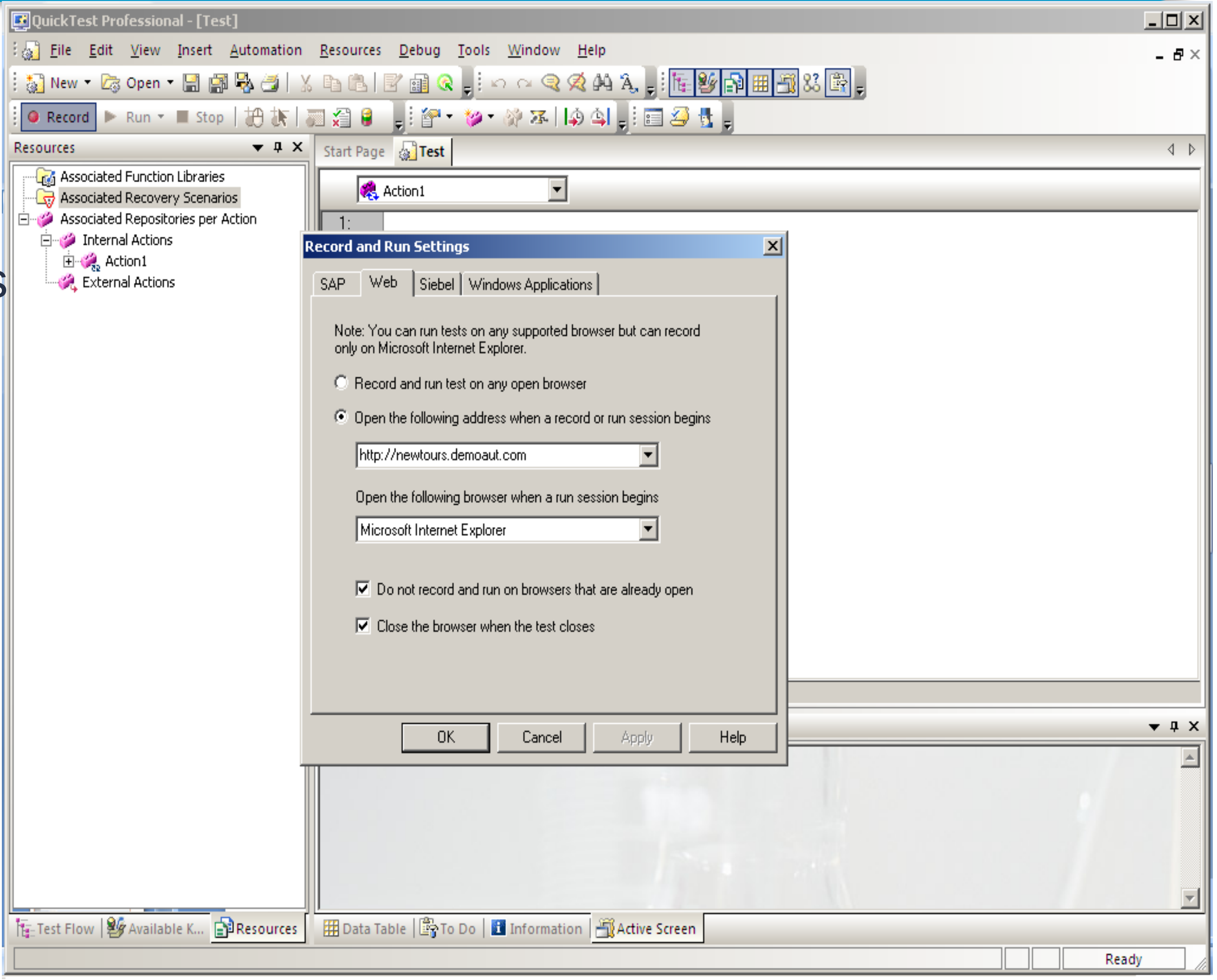
Low-Level Recording—Enables you to record on any object in your application, whether or not QTP recognizes the specific object or the specific operation. This mode records at the object level and records all run-time objects as Window or WinObject test objects. Use low-level recording for recording in an environment or on an object not recognized by QTP. You can also use low-level recording if the exact coordinates of the object are important for your test or component.

Recording a Test

1. Click on Record(F3) button to start the recording. It launches Record and Run Settings which has settings for each selected add-ins.

2. Perform certain steps

3. Click Stop(F4) button to stop the recording


QuickTest Professional - [Test]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Resources
Associated Function Libraries
Associated Recovery Scenarios
Associated Repositories per Action
Internal Actions
Action1
External Actions
Start Page Test
Action1
Test Flow Available K... Resources
Data Table To Do Information Active Screen
Ready

Record and Run Settings

SAP | Web | Siebel | Windows Applications

Note: You can run tests on any supported browser but can record only on Microsoft Internet Explorer.

( ) Record and run test on any open browser

(•) Open the following address when a record or run session begins

http://newtours.demoaut.com

Open the following browser when a run session begins

Microsoft Internet Explorer

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

[✓] Close the browser when the test closes

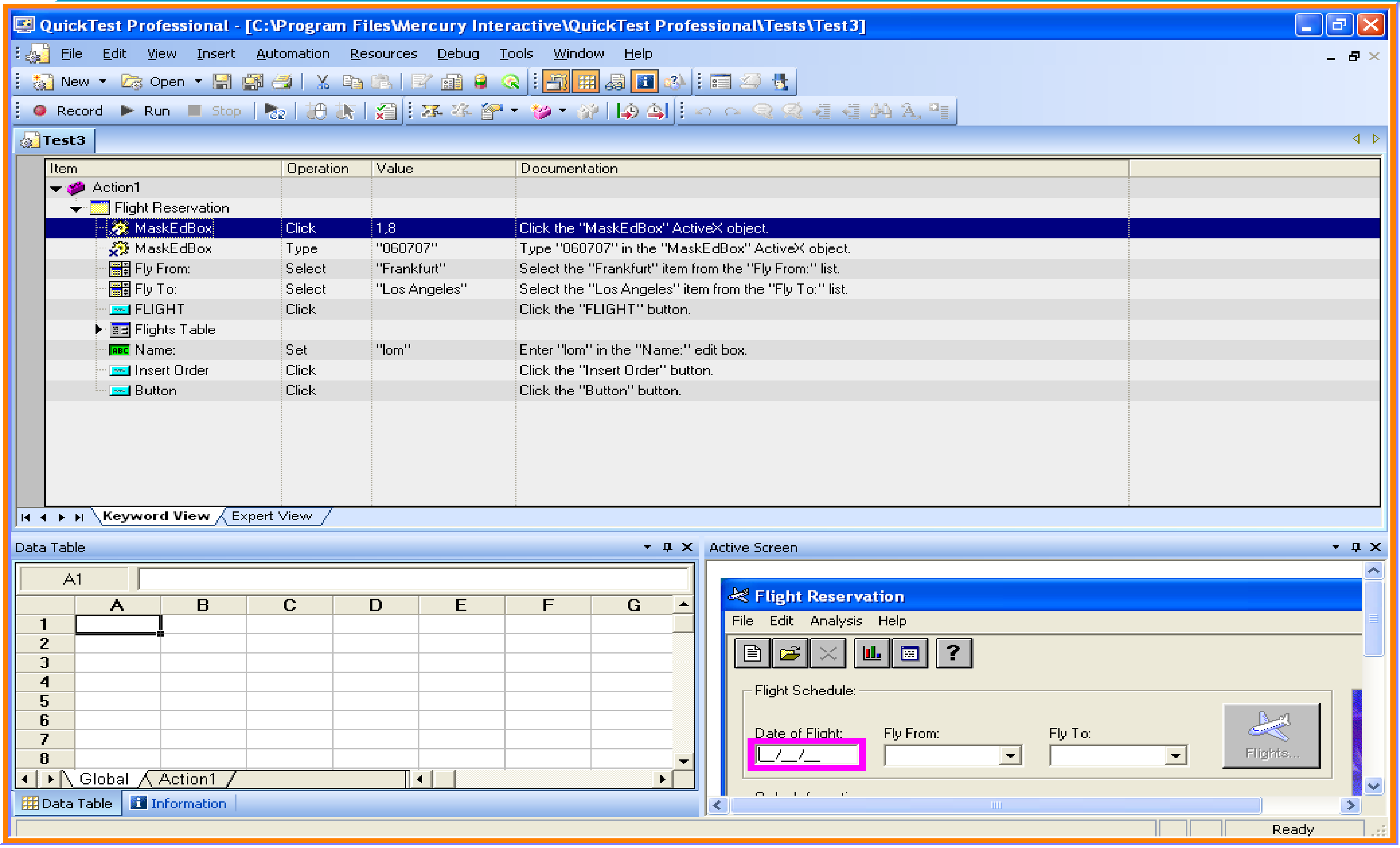
OK | Cancel | Apply | Help

Views of the scripts

The recorded test is displayed in two views :

1. ***Keyword View*** which shows the test in a keyword driven way, which is modular and has documentation to explain each step explicitly.
2. ***Expert View*** shows the underlying VB script code corresponding to each of the operation performed while recording.

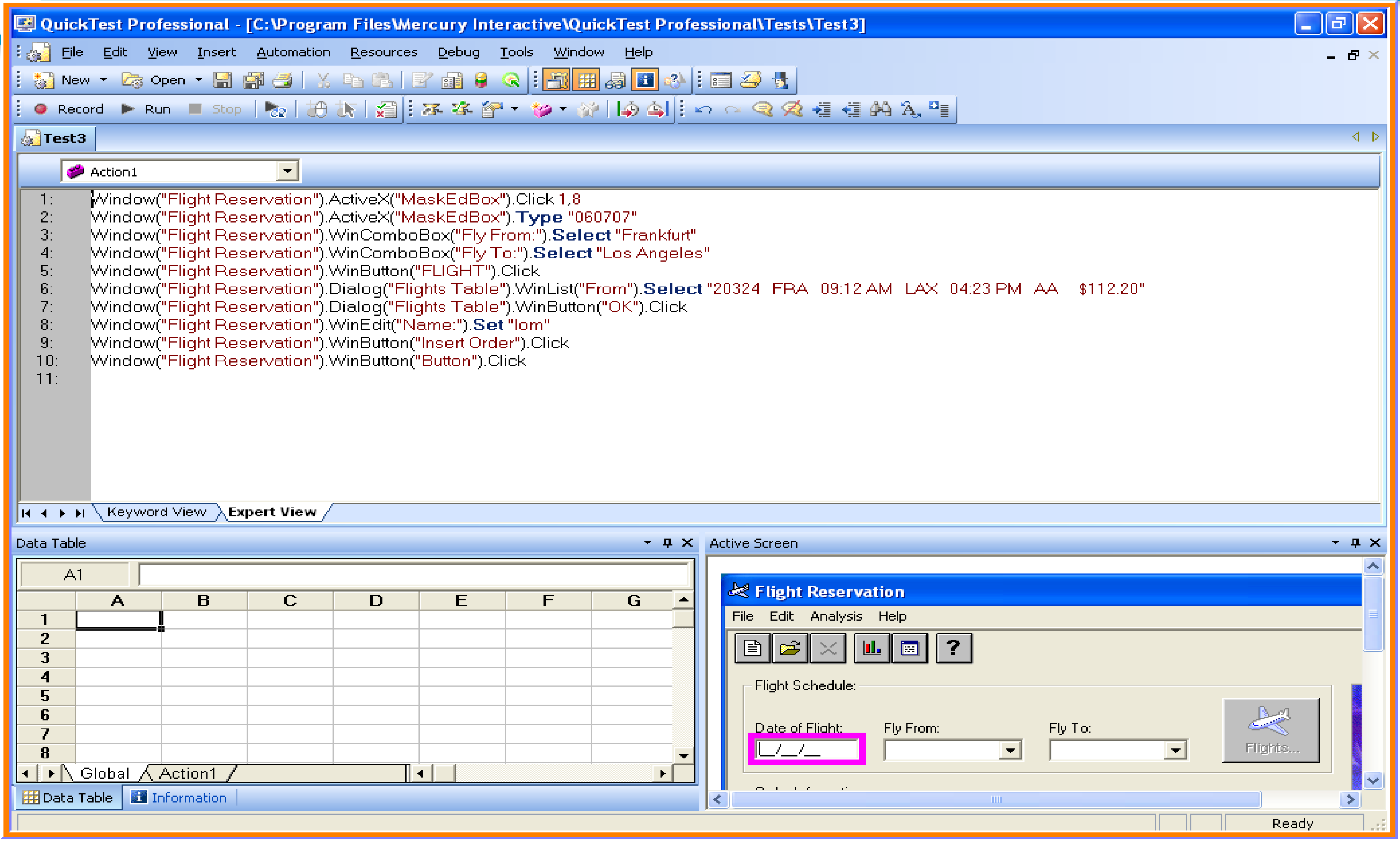
Keyword view of the recorded script



QuickTest Professional - [C:\Program Files\Mercury Interactive\QuickTest Professional\Tests\Test3]

Item	Operation	Value	Documentation
Action1			
Flight Reservation			
MaskEdBox	Click	1,8	Click the "MaskEdBox" ActiveX object.
MaskEdBox	Type	"060707"	Type "060707" in the "MaskEdBox" ActiveX object.
Fly From:	Select	"Frankfurt"	Select the "Frankfurt" item from the "Fly From:" list.
Fly To:	Select	"Los Angeles"	Select the "Los Angeles" item from the "Fly To:" list.
FLIGHT	Click		Click the "FLIGHT" button.
Flights Table			
Name:	Set	"lom"	Enter "lom" in the "Name:" edit box.
Insert Order	Click		Click the "Insert Order" button.
Button	Click		Click the "Button" button.

Expert View of the recorded script

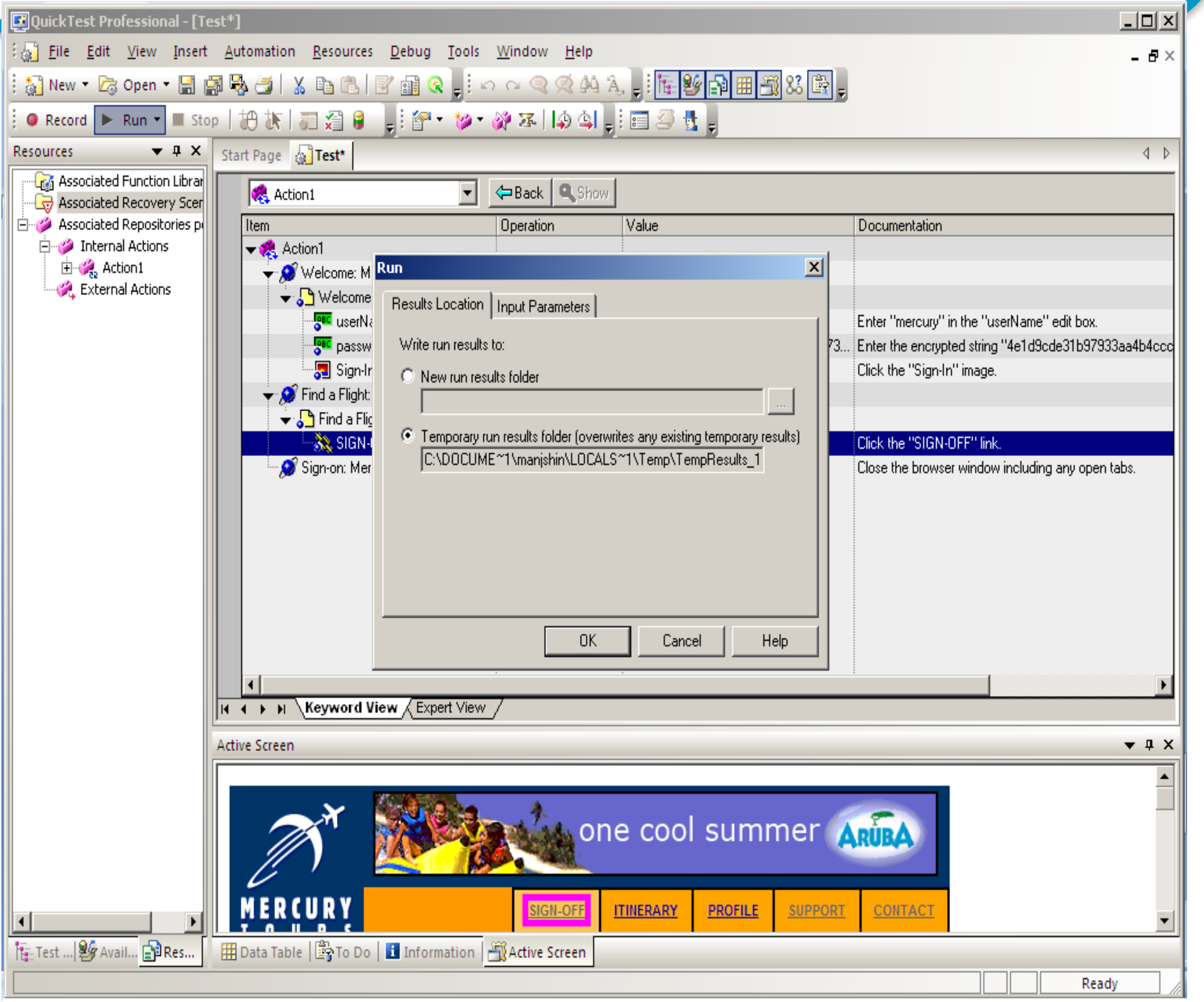


```
Window("Flight Reservation").ActiveX("MaskEdBox").Click 1,8
Window("Flight Reservation").ActiveX("MaskEdBox").Type "060707"
Window("Flight Reservation").WinComboBox("Fly From:").Select "Frankfurt"
Window("Flight Reservation").WinComboBox("Fly To:").Select "Los Angeles"
Window("Flight Reservation").WinButton("FLIGHT").Click
Window("Flight Reservation").Dialog("Flights Table").WinList("From").Select "20324   FRA   09:12 AM   LAX   04:23 PM   AA     $112.20"
Window("Flight Reservation").Dialog("Flights Table").WinButton("OK").Click
Window("Flight Reservation").WinEdit("Name:").Set "lom"
Window("Flight Reservation").WinButton("Insert Order").Click
Window("Flight Reservation").WinButton("Button").Click
```

Running the test

1. Click on Run(F5) button to start the execution.

2. Run dialog is displayed


QuickTest Professional - [Test*]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Resources
Start Page Test*
Action1
Back Show
Keyword View Expert View
Active Screen
Test ... Avail... Res...
Data Table To Do Information Active Screen
Ready

Associated Function Librar
Associated Recovery Scer
Associated Repositories p
Internal Actions
Action1
External Actions

Item	Operation	Value	Documentation
Action1			
Welcome: M			
Welcome			
userNa			Enter "mercury" in the "userName" edit box.
passw		73...	Enter the encrypted string "4e1d9cde31b97933aa4b4ccc
Sign-Ir			Click the "Sign-In" image.
Find a Flight:			
Find a Flig			
SIGN-			Click the "SIGN-OFF" link.
Sign-on: Mer			Close the browser window including any open tabs.

Run

Results Location | Input Parameters

Write run results to:

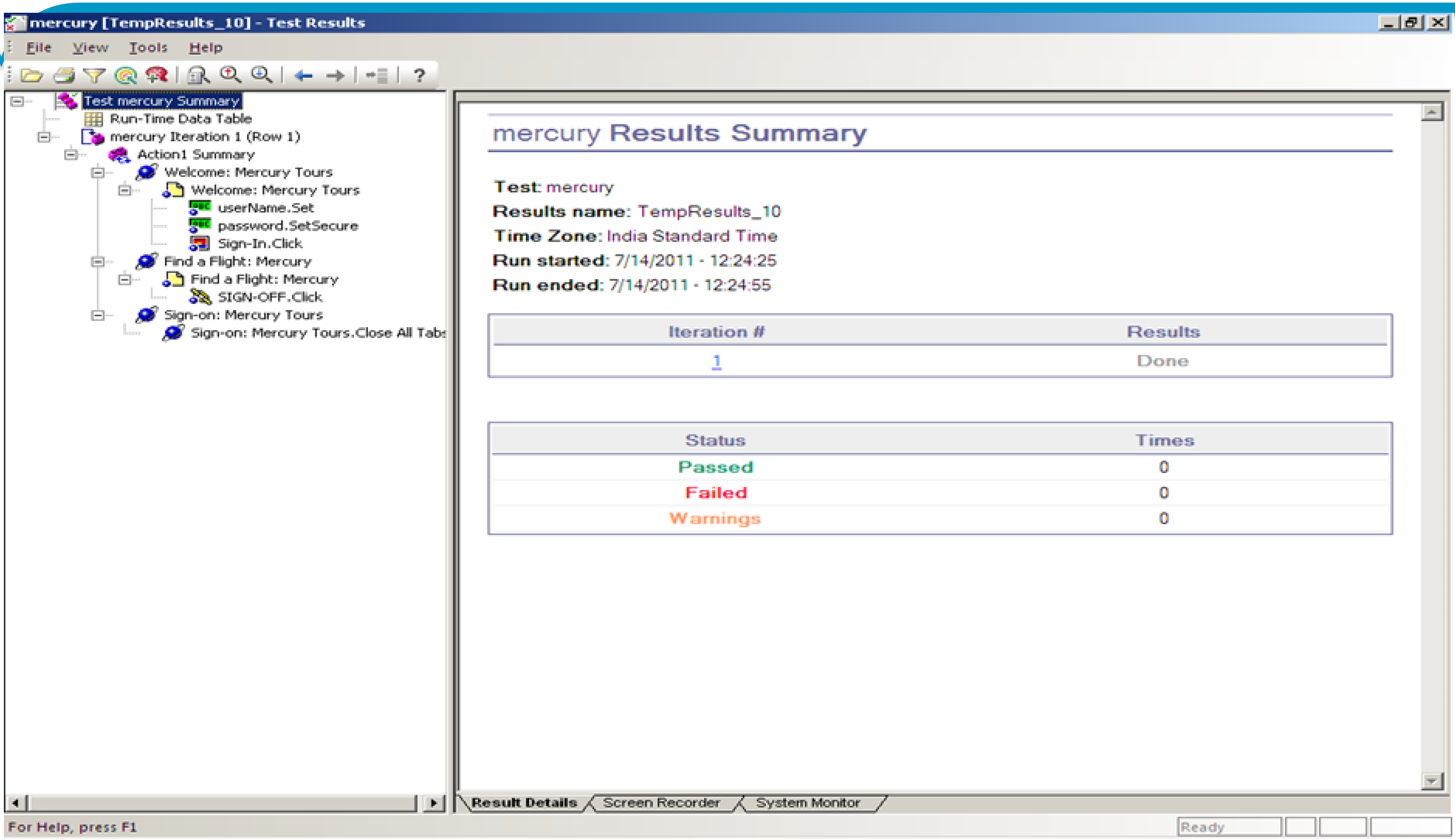
( ) New run results folder

(●) Temporary run results folder (overwrites any existing temporary results)

C:\DOCUME~1\manjshin\LOCALS~1\Temp\TempResults_1

OK Cancel Help

Test Result Window



- Test mercury Summary
  - Run-Time Data Table
  - mercury Iteration 1 (Row 1)
    - Action1 Summary
      - Welcome: Mercury Tours
        - Welcome: Mercury Tours
          - userName.Set
          - password.SetSecure
          - Sign-In.Click
      - Find a Flight: Mercury
        - Find a Flight: Mercury
          - SIGN-OFF.Click
      - Sign-on: Mercury Tours
        - Sign-on: Mercury Tours.Close All Tabs

mercury Results Summary

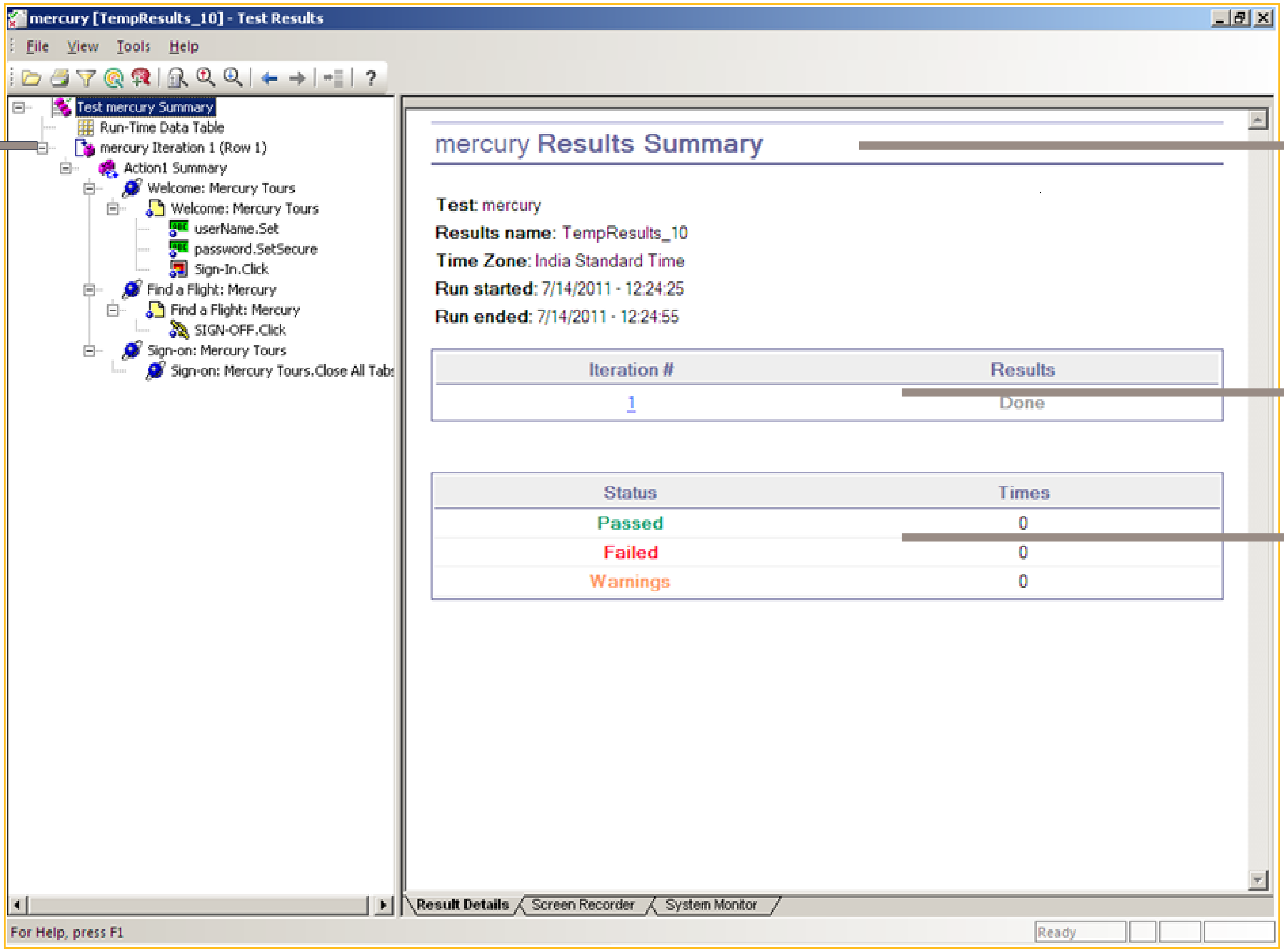
Test: mercury
Results name: TempResults_10
Time Zone: India Standard Time
Run started: 7/14/2011 - 12:24:25
Run ended: 7/14/2011 - 12:24:55

Iteration #	Results
1	Done

Status	Times
Passed	0
Failed	0
Warnings	0

Test Results Pane

Test Result Tree



mercury Results Summary

Test: mercury
Results name: TempResults_10
Time Zone: India Standard Time
Run started: 7/14/2011 - 12:24:25
Run ended: 7/14/2011 - 12:24:55

Iteration #	Results
1	Done

Status	Times
Passed	0
Failed	0
Warnings	0

Test Result Details

Iteration Summary

Status Summary

Parameterization

Parameterization allows creating maintainable scripts which can run with different set of data.

The data is not hard-coded in the script and are replaced with parameters in order to execute the script with different data.

Types of Parameterization

1. ***Data Table parameters*** → It helps to create a data-driven test (or action) that runs several times using the data in the data sheet. QTP substitutes the constant value with a different value from the Data Table for each iteration.
2. ***Environment Variables*** → It helps in using data from other sources like environment variables.
3. ***Random Number variables*** → The random number input generates random numbers and uses them as input value for a parameter. By default, the random number ranges between 0 and 100. A different random number is generated every time the parameter is called for every iteration or for every Test run.
4. ***Test/Action Parameter*** → Action parameters enable you to transfer input values from your test to a top-level action, from a parent action to a nested action, or from an action to a sibling action that occurs later in the test . For e.g. Msgbox Parameter("Name")

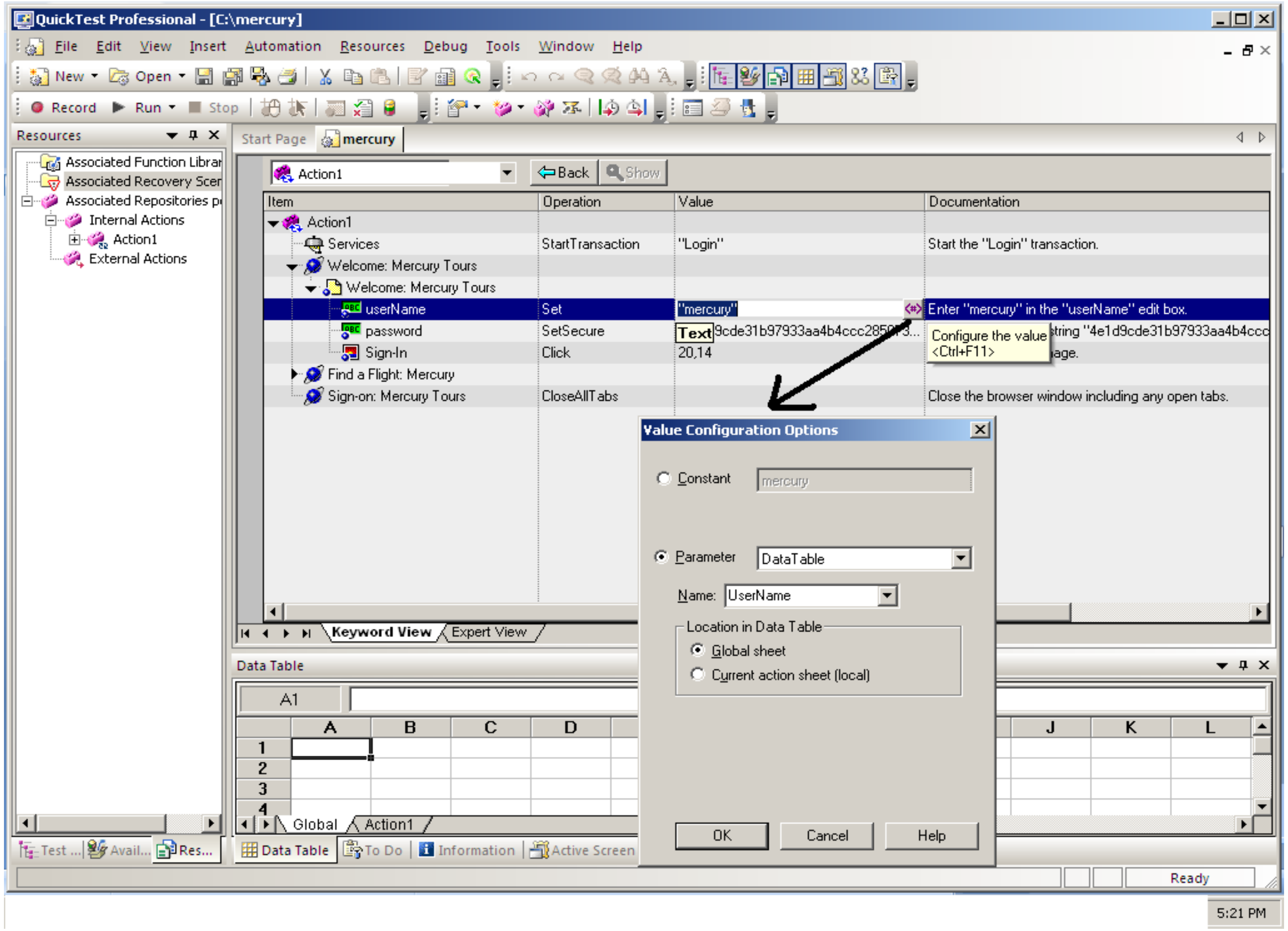
   Test Parameter is same as action parameter only the difference is that it will be available to all the actions in the test in which it is defined.

   For e.g. TestArgs("Name") = Value

Parameterizing in Global sheet of DataTable

Code generated in Expert View:

```
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").Set DataTable("UserName", dtGlobalSheet)
```


QuickTest Professional - [C:\mercury]

Action1

Item	Operation	Value	Documentation
Action1			
Services	StartTransaction	"Login"	Start the "Login" transaction.
Welcome: Mercury Tours			
Welcome: Mercury Tours			
userName	Set	"mercury"	Enter "mercury" in the "userName" edit box.
password	SetSecure	Text 9cde31b97933aa4b4ccc2850f3...	tring "4e1d9cde31b97933aa4b4ccc
Sign-In	Click	20,14	age.
Find a Flight: Mercury			
Sign-on: Mercury Tours	CloseAllTabs		Close the browser window including any open tabs.

Configure the value <Ctrl+F11>

Value Configuration Options

Constant: mercury

[✓ Parameter]: DataTable

Name: UserName

Location in Data Table: [✓ Global sheet] Current action sheet (local)

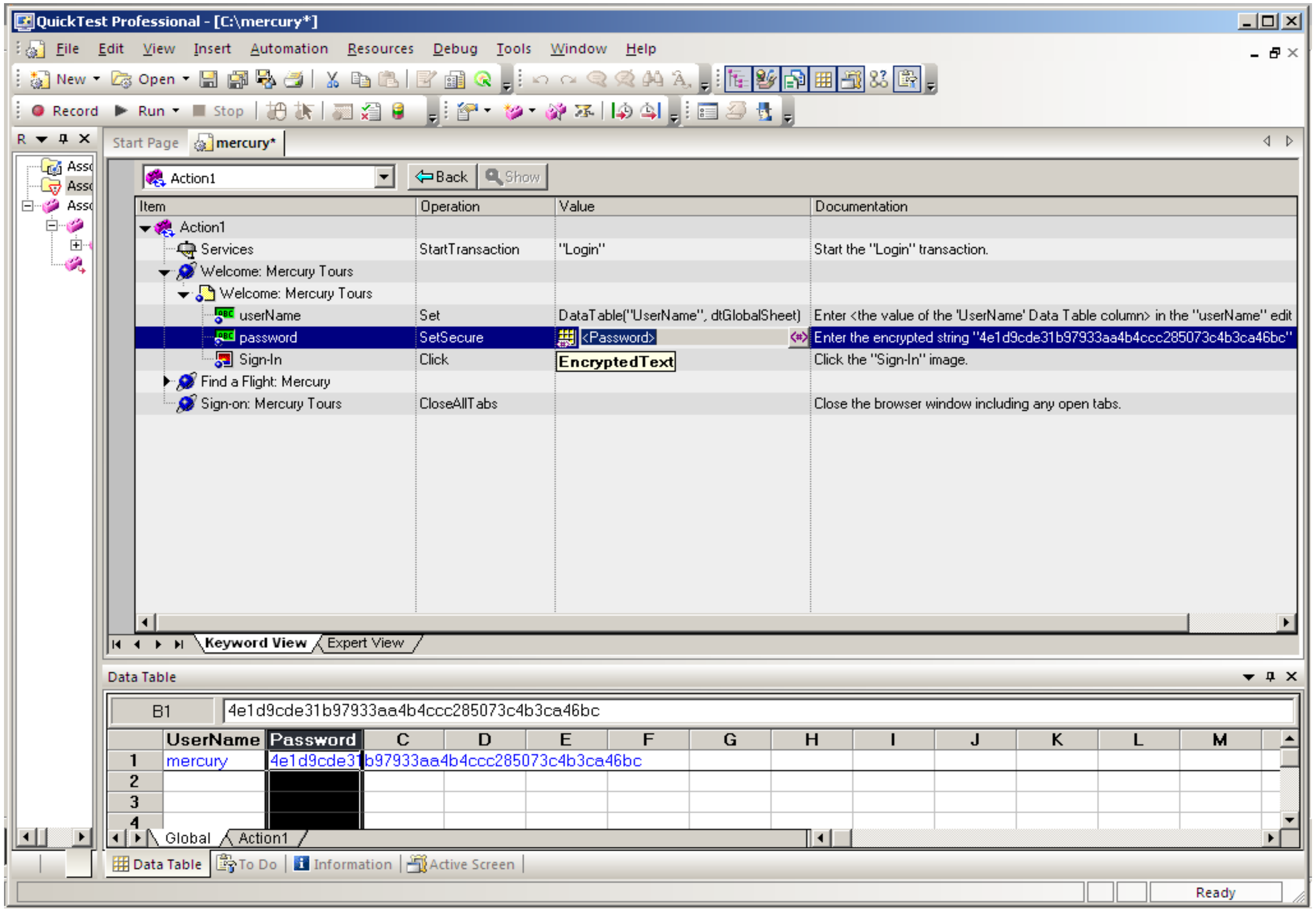
OK Cancel Help

Parameterizing in Global and Local sheet of DataTable

Parameterized both the hard coded values in scripts

For e.g. Username and Password data is parameterized and derived from Global sheet.

Similarly it can be derived from Local sheet too.


QuickTest Professional - [C:\mercury*]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open Record Run Stop
Start Page mercury*
Action1 Back Show
Keyword View Expert View
Global Action1
Data Table To Do Information Active Screen
Ready

Item	Operation	Value	Documentation
Action1			
Services	StartTransaction	"Login"	Start the "Login" transaction.
Welcome: Mercury Tours			
Welcome: Mercury Tours			
userName	Set	DataTable("UserName", dtGlobalSheet)	Enter <the value of the 'UserName' Data Table column> in the "userName" edit
password	SetSecure	<Password>	Enter the encrypted string "4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc"
Sign-In	Click	EncryptedText	Click the "Sign-In" image.
Find a Flight: Mercury			
Sign-on: Mercury Tours	CloseAllTabs		Close the browser window including any open tabs.

Data Table

B1 4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc

	UserName	Password	C	D	E	F	G	H	I	J	K	L	M
1	mercury	4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc											
2													
3													
4													

Environment Variables

Types of Environment Variables

Built-in – These are provided by QTP. Built-in variable list can be accessed from File->Settings->Environment option.

E.g. sOS = Environment("OS")

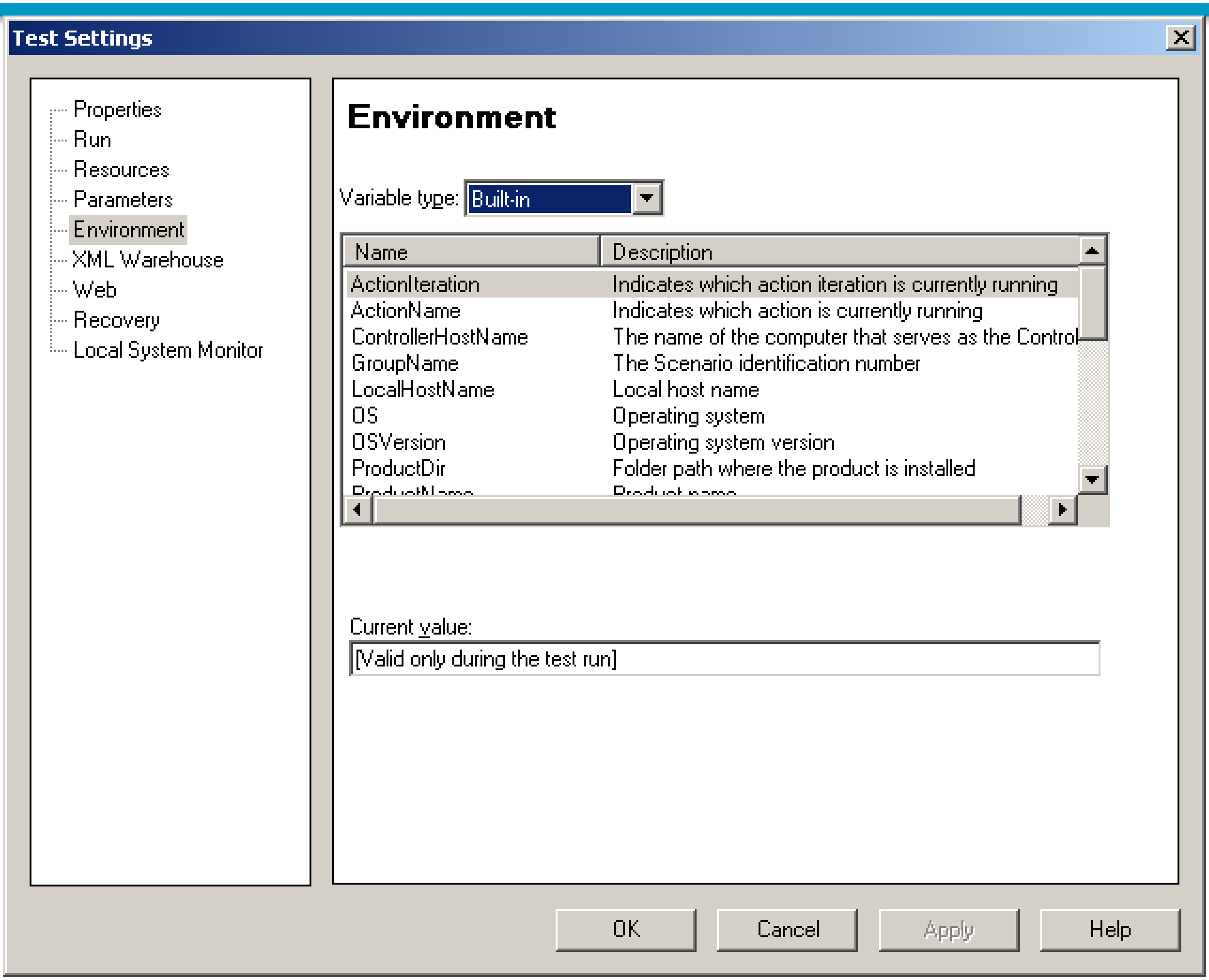
User Defined – Tester can define own environment variables. There are two types of User defined variables

- Internal: These are the variables that are defined in the test.

  E.g. Environment.Value("Path") = "C:\Test"

- External : These are the variables that predefined in the external environment file (xml) and can be used in the test

Environment variables on Test Settings dialog



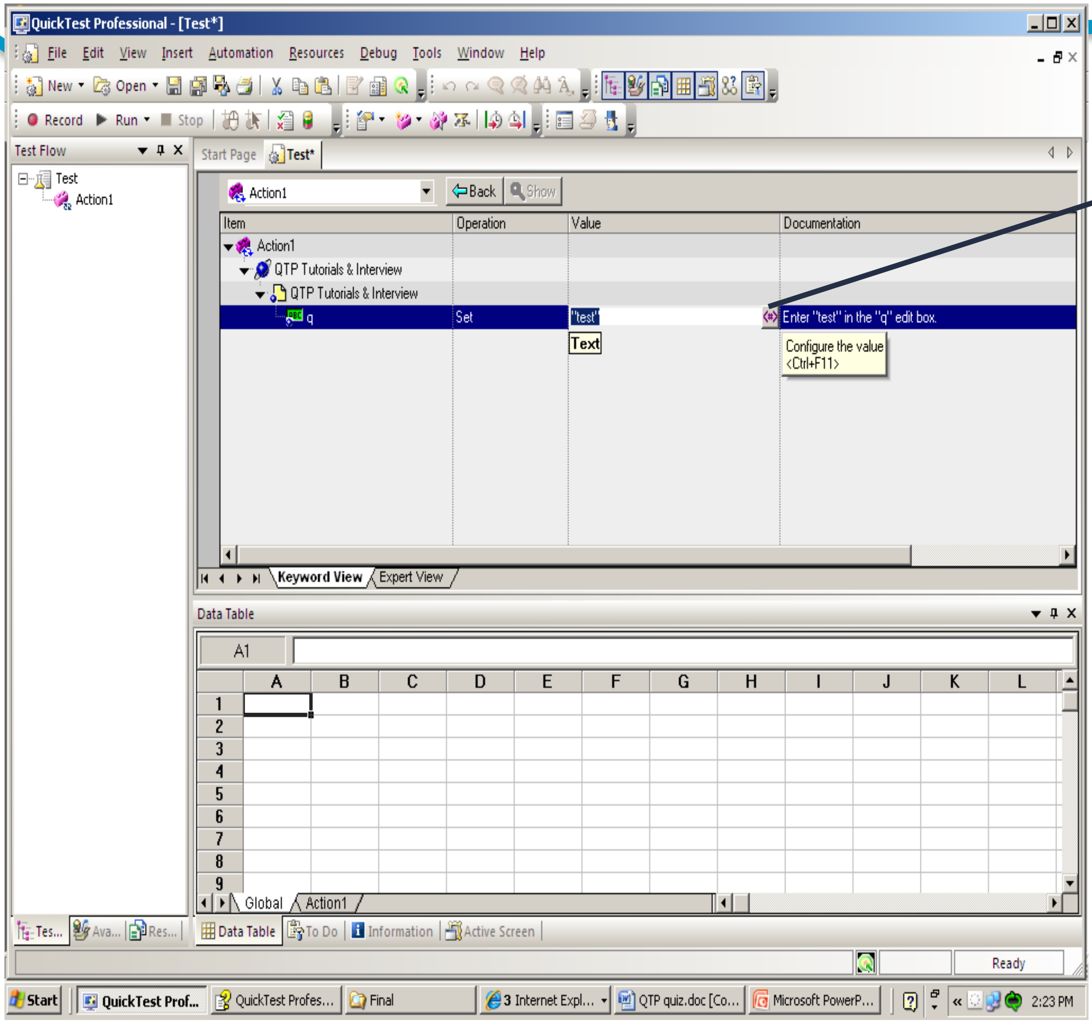
Environment

Variable type: Built-in

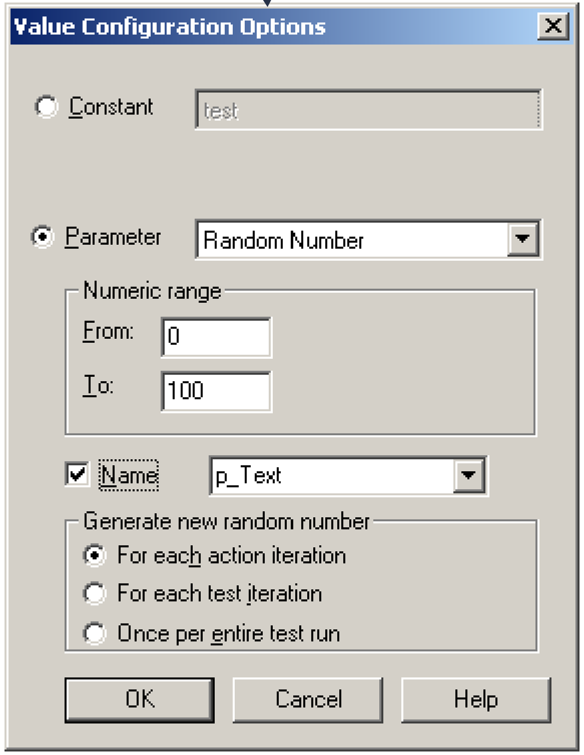
Name	Description
ActionIteration	Indicates which action iteration is currently running
ActionName	Indicates which action is currently running
ControllerHostName	The name of the computer that serves as the Control
GroupName	The Scenario identification number
LocalHostName	Local host name
OS	Operating system
OSVersion	Operating system version
ProductDir	Folder path where the product is installed
ProductName	Product name

Current value: [Valid only during the test run]

Random numbers


QuickTest Professional - [Test*]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open Record Run Stop
Test Flow
Test
Action1
Start Page Test*
Action1 Back Show
Item Operation Value Documentation
Action1
QTP Tutorials & Interview
QTP Tutorials & Interview
q Set "test" Enter "test" in the "q" edit box.
Text
Configure the value <Ctrl+F11>
Keyword View Expert View
Data Table
A1
Global Action1
Data Table To Do Information Active Screen
Ready
Start QuickTest Prof... QuickTest Profes... Final 3 Internet Expl... QTP quiz.doc [Co... Microsoft PowerP... 2:23 PM

Clicking on 'Configure the value' shows 'Value Configuration options' dialog



Value Configuration Options

Constant: test

[✓ Parameter]: Random Number

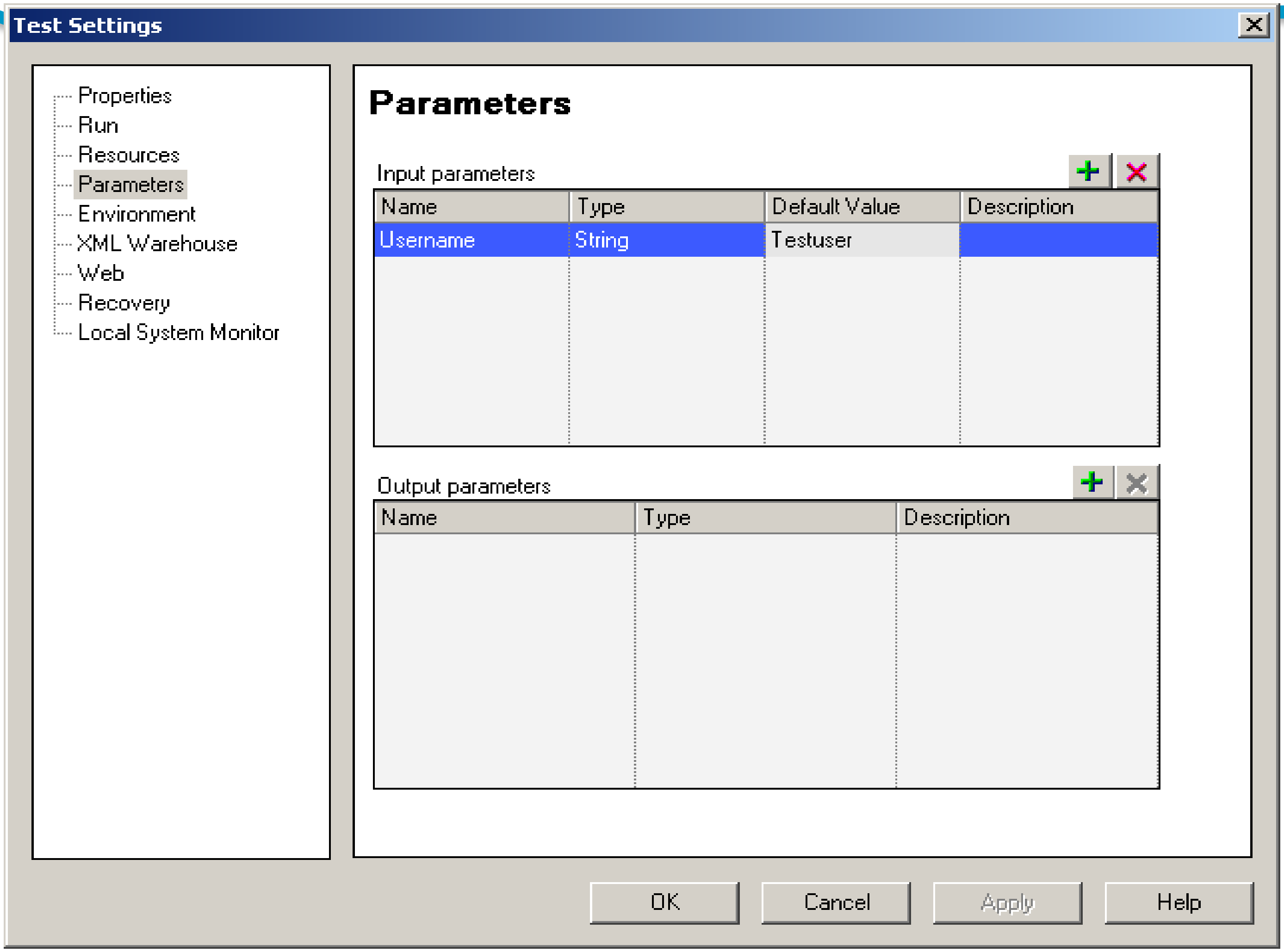
Numeric range
- From: 0
- To: 100

[✓ Name]: p_Text

Generate new random number
- [✓ For each action iteration]
- For each test iteration
- Once per entire test run

OK | Cancel | Help

Test Parameters on Test Settings dialog



Name	Type	Default Value	Description
Username	String	Testuser	

Name	Type	Description

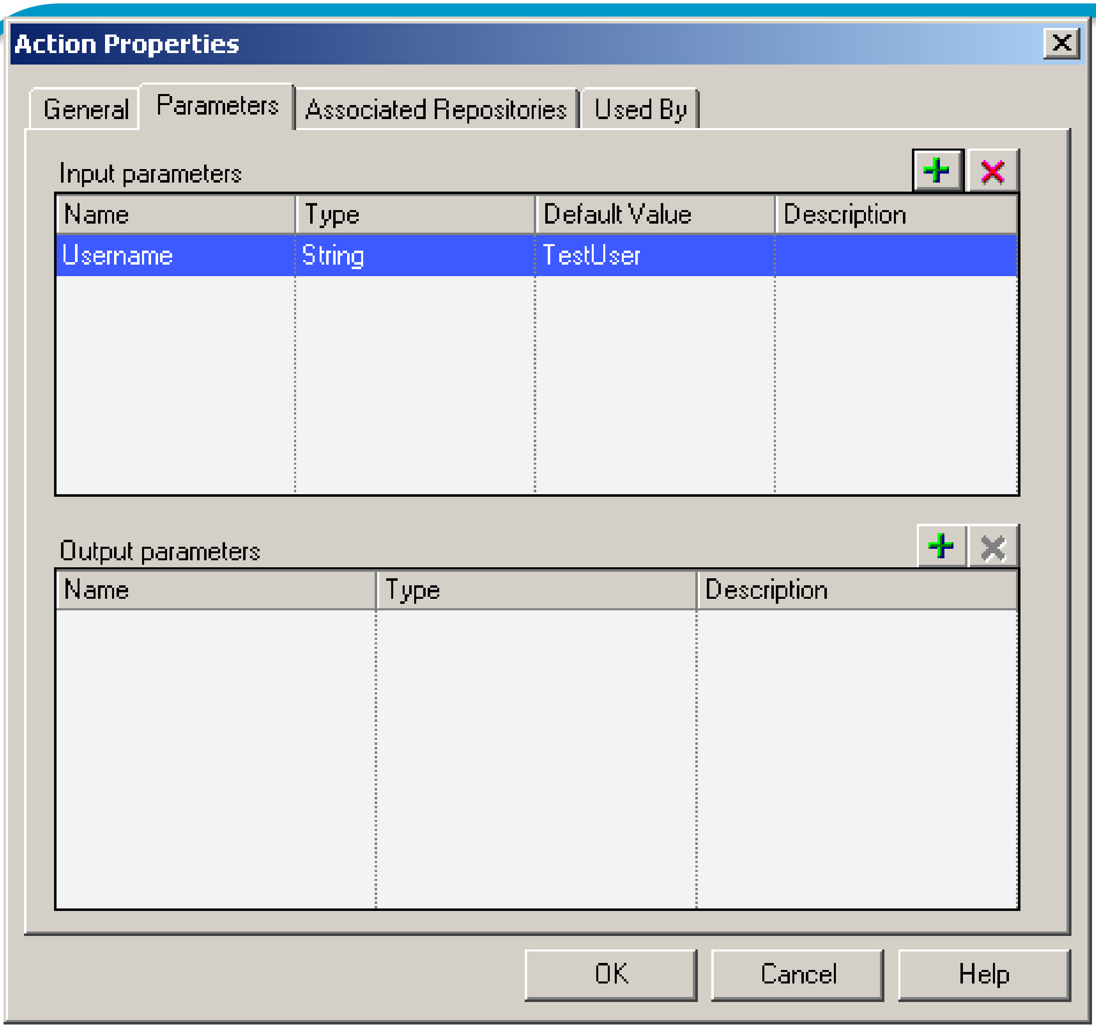
❑ Code to display the value in 'Username' Test Parameter:

```
Msgbox
TestArgs("Username")
```

❑ Code to change the Test Parameter:

```
TestArgs("Username")
= "newname"
```

Action Parameters on Action Properties dialog



Input parameters

Name	Type	Default Value	Description
Username	String	TestUser	

Output parameters

Name	Type	Description

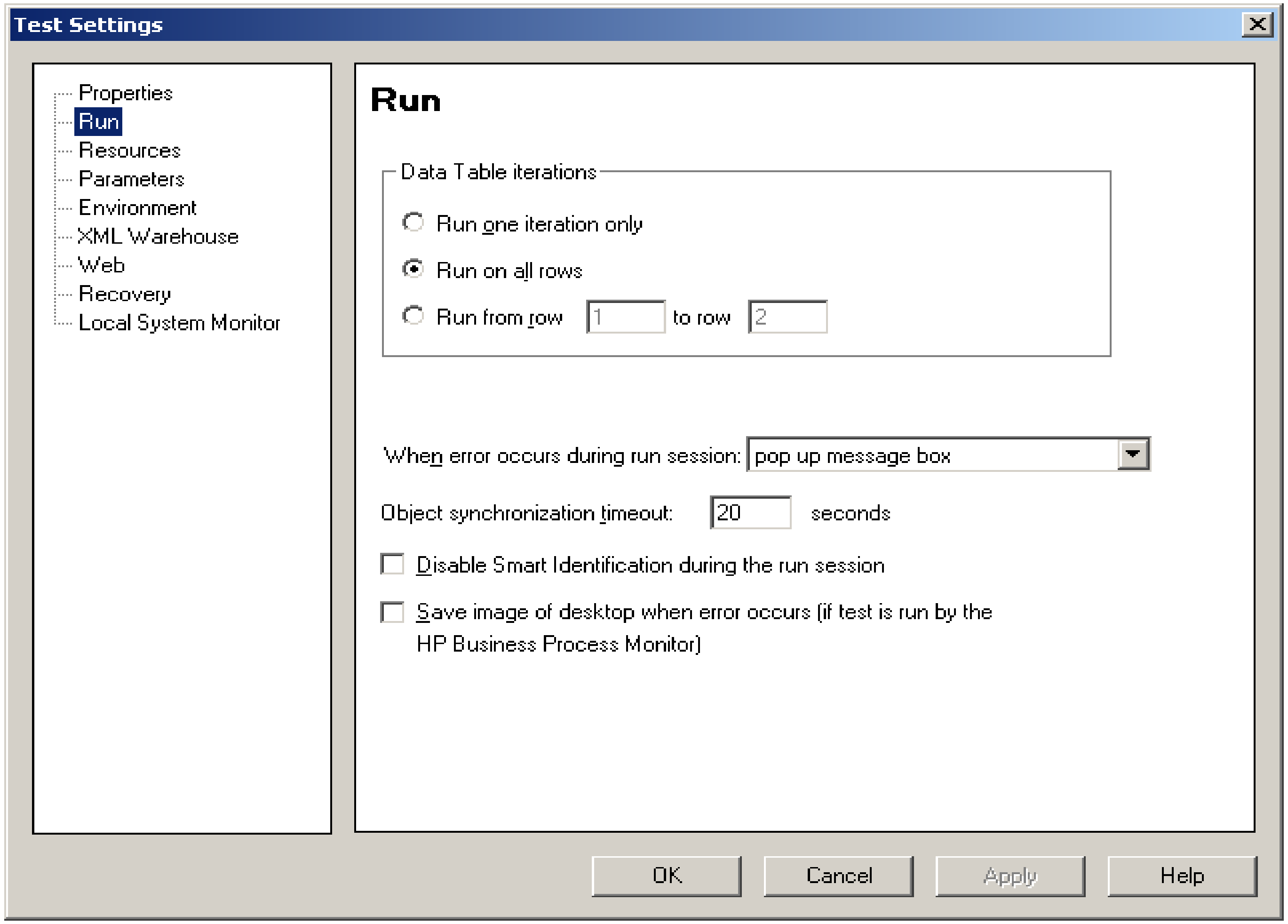
- ❑ Code to display the value in ‘Username’ Action Parameter:

```
Msgbox Parameter("Username")
```

- ❑ Code to change the Action Parameter:

```
Parameter("Username") = “newname”
```

Test Settings for executing all the iterations



Run

Data Table iterations

( ) Run one iteration only
(●) Run on all rows
( ) Run from row [1] to row [2]

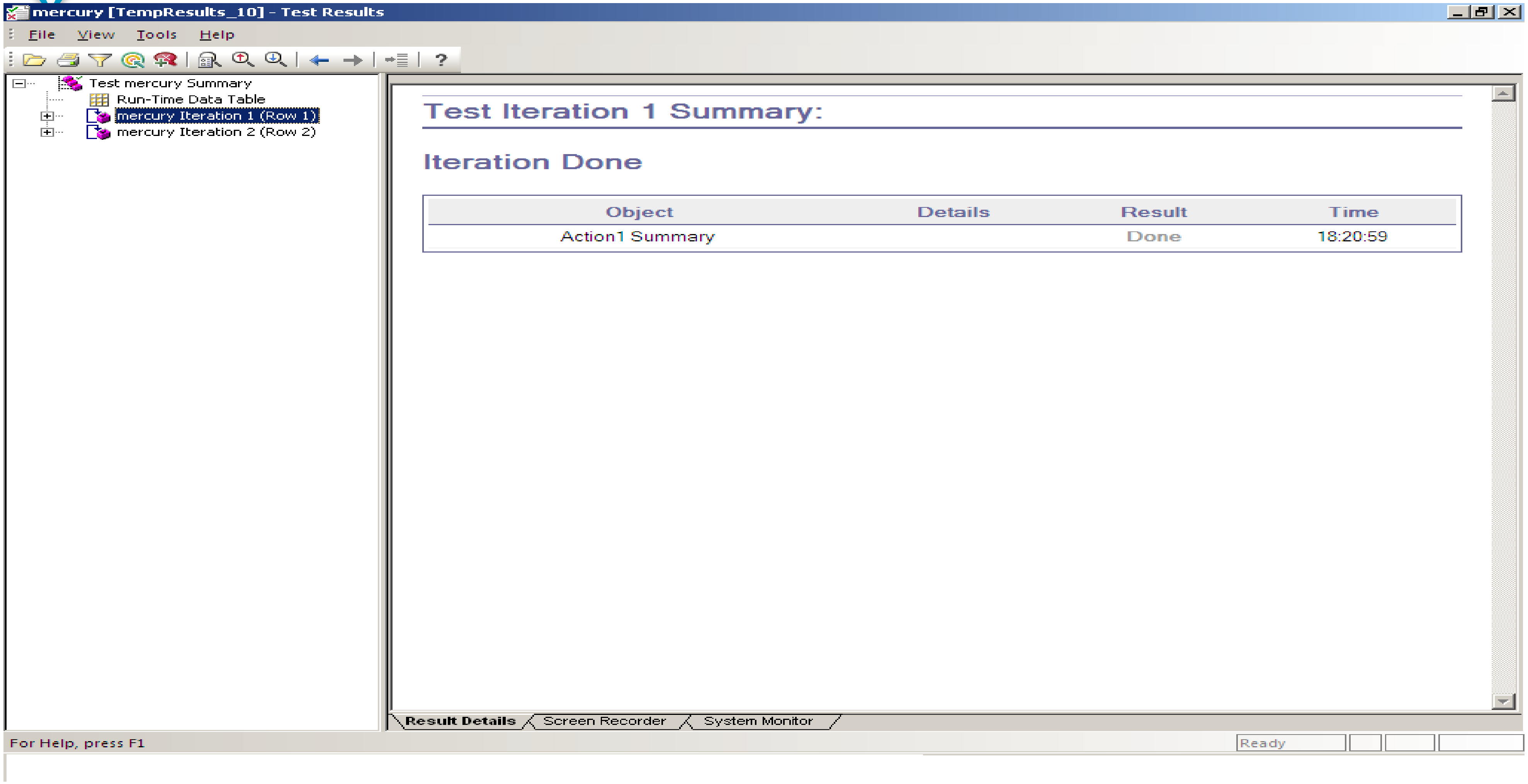
When error occurs during run session: pop up message box

Object synchronization timeout: 20 seconds

☐ Disable Smart Identification during the run session

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

Result File with multiple iterations



Test Iteration 1 Summary:

Iteration Done

Object	Details	Result	Time
Action1 Summary		Done	18:20:59

Introduction to Actions

QTP is made up of logical units called actions.

Actions help in making the test modular and increase the reusability of test script.

A given test can have multiple actions, with each action containing a logical sequence of steps.

Actions can be of three kinds. They are :

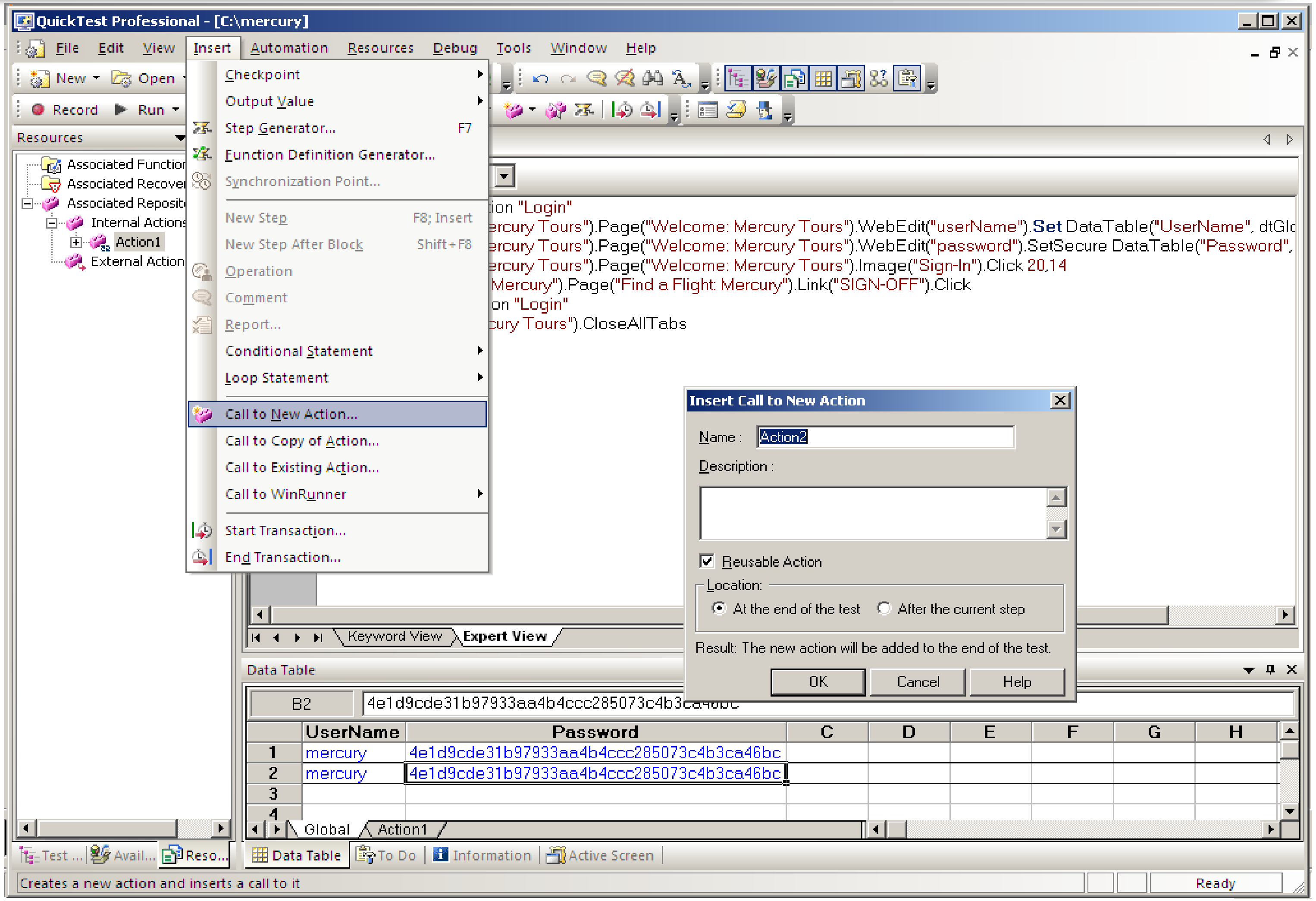
- ***Non reusable action*** : An action which cannot be called by other actions. By default all actions which get created are Non-reusable actions.
- ***Reusable actions*** : An action that can be called multiple times by the same test or any other test
- ***External action*** : A reusable action which can be called from a different test

Creating new Actions

Select Insert->Call to New Action menu

Insert Call to New Action dialog is displayed

Enter Action name and check the checkbox in case Reusable Action needs to be created



```
ion "Login"
ercury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").Set DataTable("UserName", dtGlc
ercury Tours").Page("Welcome: Mercury Tours").WebEdit("password").SetSecure DataTable("Password",
ercury Tours").Page("Welcome: Mercury Tours").Image("Sign-In").Click 20,14
Mercury").Page("Find a Flight: Mercury").Link("SIGN-OFF").Click
on "Login"
cury Tours").CloseAllTabs
```

Insert Call to New Action

Name : Action2

Description :

☑ Reusable Action

Location:

(● At the end of the test) ○ After the current step

Result: The new action will be added to the end of the test.

OK | Cancel | Help

B2 | 4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc

	UserName	Password	C	D	E	F	G	H
1	mercury	4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc						
2	mercury	4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc						
3								
4								

Creates a new action and inserts a call to it | Ready

Action call properties

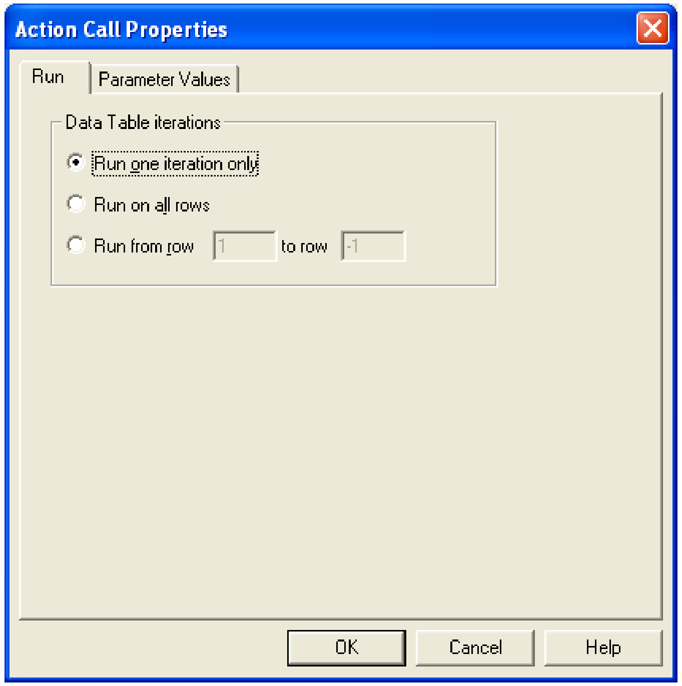
Action call properties determine the number of iterations the called action can perform. The properties set is applicable only to particular action call.

The options available are as follows:

Run one iteration only→ Runs the called action only once

Run on all rows→ Runs the called action with the number of iterations according to the number of rows in the action's Data Table.

Run from row __ to row __→ Runs the called action with the number of iterations according to the specified row range.

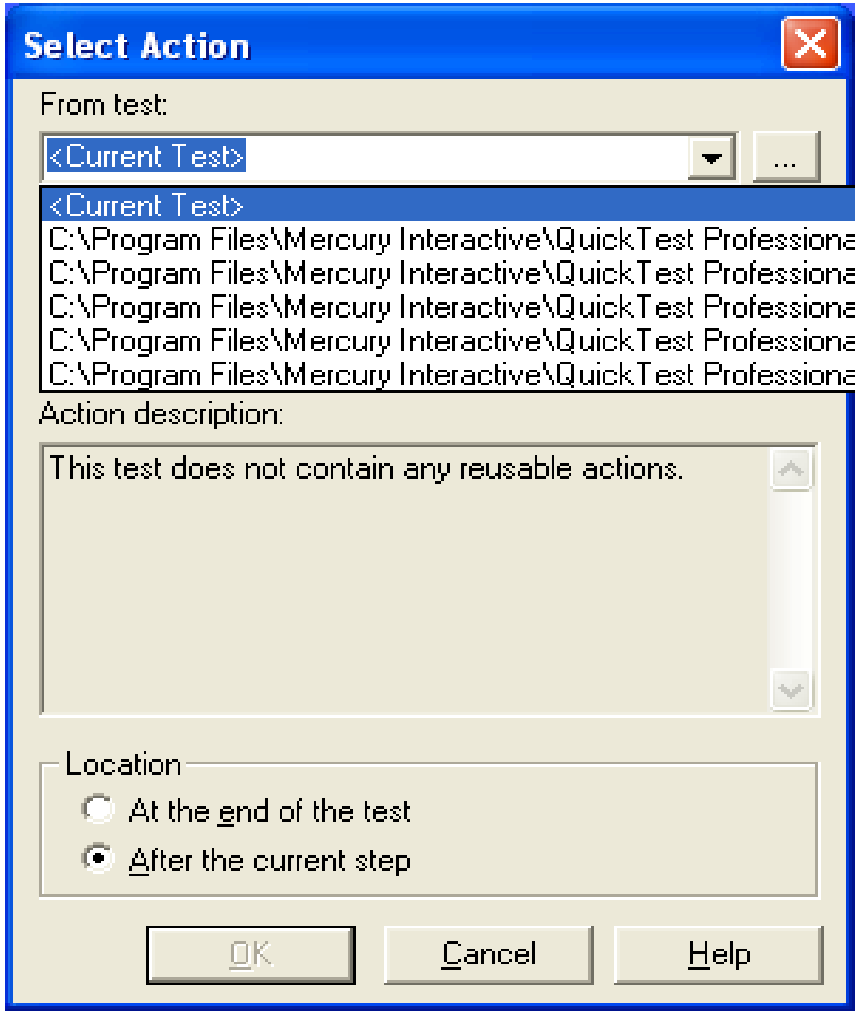


(● Run one iteration only) ○ Run on all rows ○ Run from row 1 to row -1

Inserting Calls to Existing Action

The Steps to call an external action are as follows :

- Go to Insert → Call to existing action, this opens a Select Action dialog where we can select the test from which the action needs to be called.
- Once the test is selected the reusable actions available in that test are visible. Select the Action which is required.
- Code in Expert View:
- ***RunAction "Action1", oneIteration***



<Current Test>
C:\Program Files\Mercury Interactive\QuickTest Professiona
C:\Program Files\Mercury Interactive\QuickTest Professiona
C:\Program Files\Mercury Interactive\QuickTest Professiona
C:\Program Files\Mercury Interactive\QuickTest Professiona
C:\Program Files\Mercury Interactive\QuickTest Professiona

Action description:

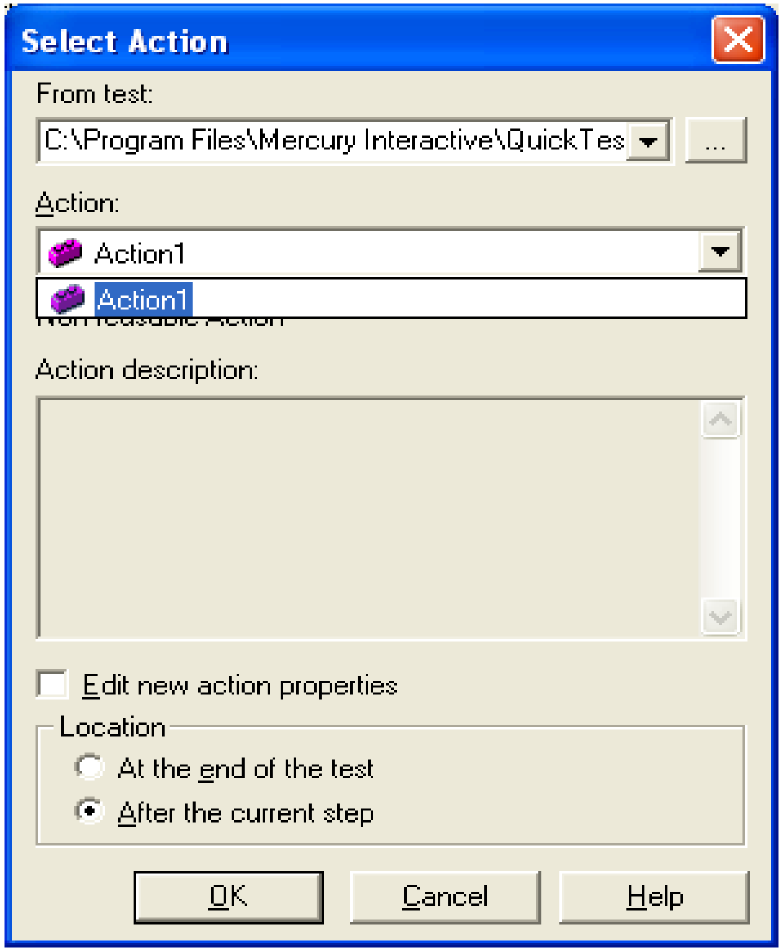
This test does not contain any reusable actions.

Location
- ( ) At the end of the test
- (●) After the current step

Inserting Calls to Copy of Action

The following steps are used for inserting calls to copies of Actions:

- Go to Insert → Call to Copy of Action, this opens a selection dialog where we can select the test from which the action needs to be called.
- Once the test is selected all the actions available in that test are visible. Select the Action which is required. Choose where to put the action.
- The called action can be modified in the calling test without affecting the actual action.



Objects and Object Identification

Objects are a representation of every item found in an application.

Objects are visual(e.g. Button and Text) and non-visual (e.g. Dictionary, Reporter) elements. Each object has its properties and methods.

Quick Test learns objects of the application based on their properties.

QTP stores the object data along with properties in the object repository.

Object Identification:

QTP uses default Object Identification properties : Mandatory & Assistive to learn objects stored in Object Repository.

If successful identification was not possible then go for Smart Identification using Base filter properties & Optional base filter properties.

Ordinal Identifiers like (index , location or creation time) can be used if Smart Identification is disabled.

Introduction of Object Repository

Object repository is the collection of the objects which QTP has identified during the recording.

The objects are represented in a tree view.

The object repository can be accessed by the following path:

- Resources → Object Repository Manager
- Click on the object repository toolbar button
- Keyboard keys (Ctrl+R)

Even when steps containing a test object are deleted from the test or component, the objects remain in the object repository.

Types of Object Repository

Local Object Repository

>QuickTest uses a separate object repository for each action.

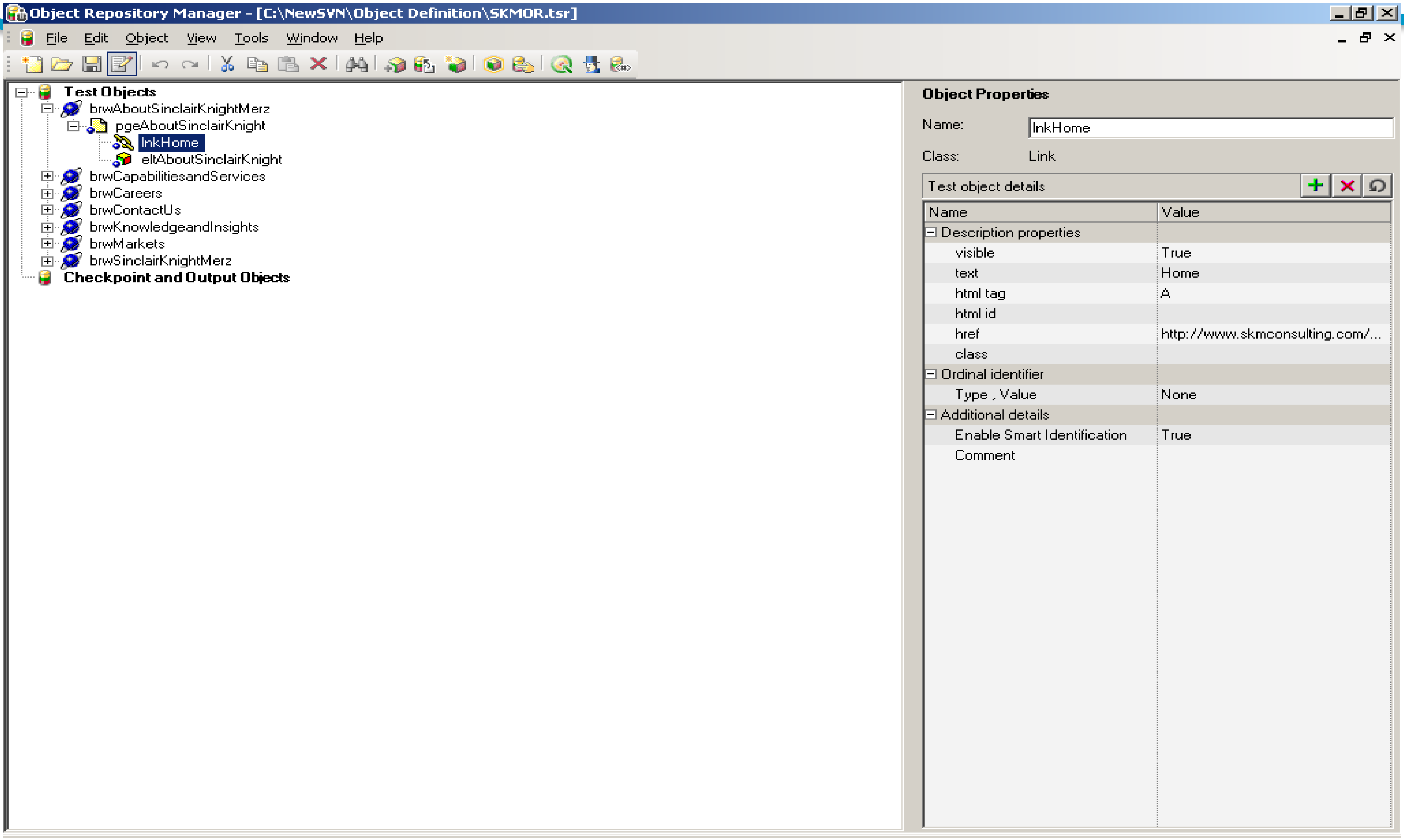
>When you save your test, all of the local object repositories are automatically saved with the test (as part of each action within the test).

Shared Object Repository

>QuickTest uses the same object repository for different actions.

>You can export the local objects to a shared object repository .

Test Objects in Object Repository


Object Repository Manager - [C:\NewSVN\Object Definition\SKMOR.tsr]

- Test Objects
  - brwAboutSinclairKnightMerz
    - pgeAboutSinclairKnight
      - lnkHome
      - eltAboutSinclairKnight
  - brwCapabilitiesandServices
  - brwCareers
  - brwContactUs
  - brwKnowledgeandInsights
  - brwMarkets
  - brwSinclairKnightMerz
- Checkpoint and Output Objects

Object Properties

Name: lnkHome

Class: Link

Test object details

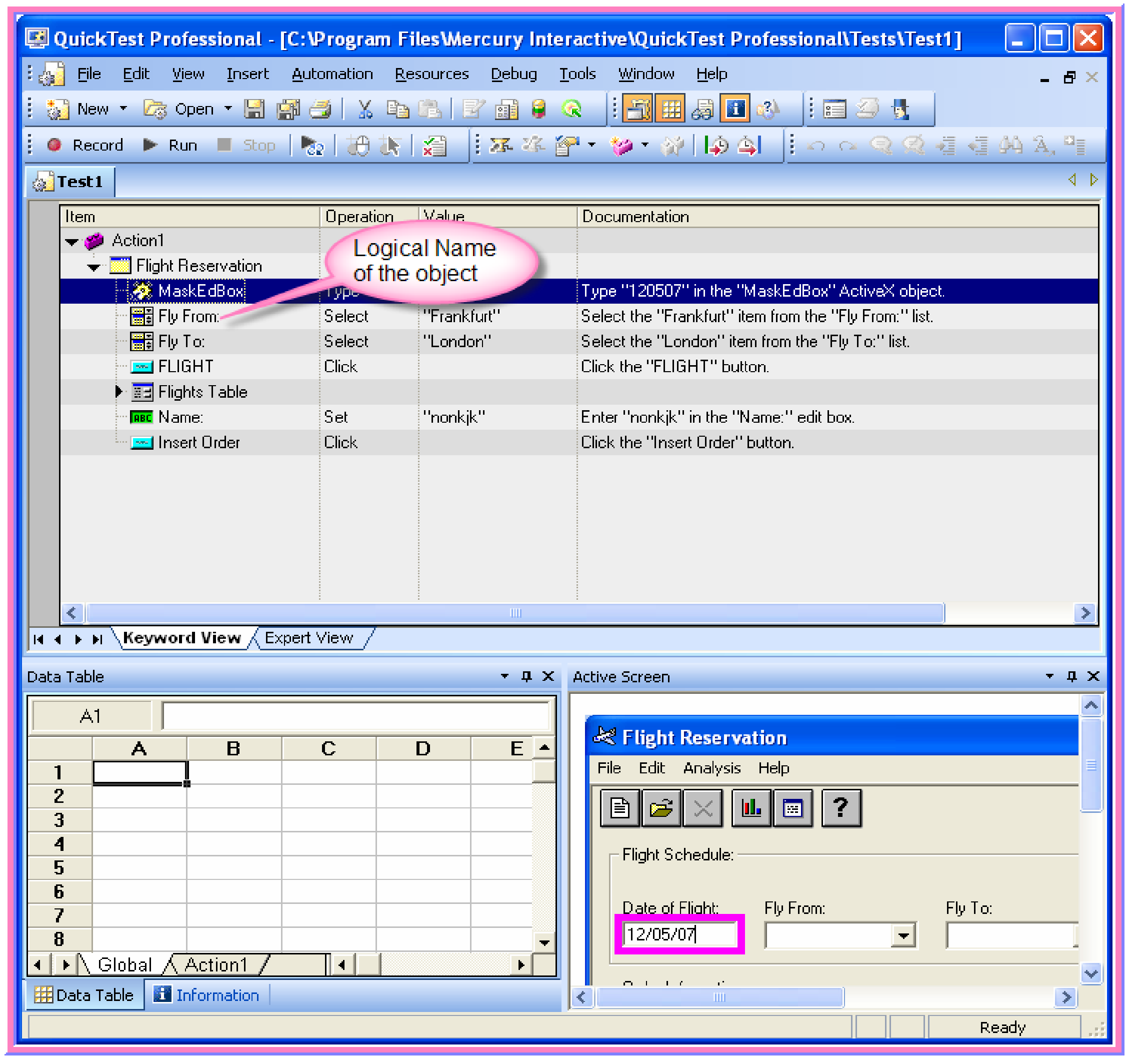
Name	Value
Description properties	
visible	True
text	Home
html tag	A
html id	
href	http://www.skmconsulting.com/...
class	
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

Logical Name of an Object

After reading the class and properties of an object, QTP assigns a logical name to the object

QTP refers to the recorded objects by using its logical name

The logical name can be edited and used in QTP scripts


QuickTest Professional - [C:\Program Files\Mercury Interactive\QuickTest Professional\Tests\Test1]

Test1

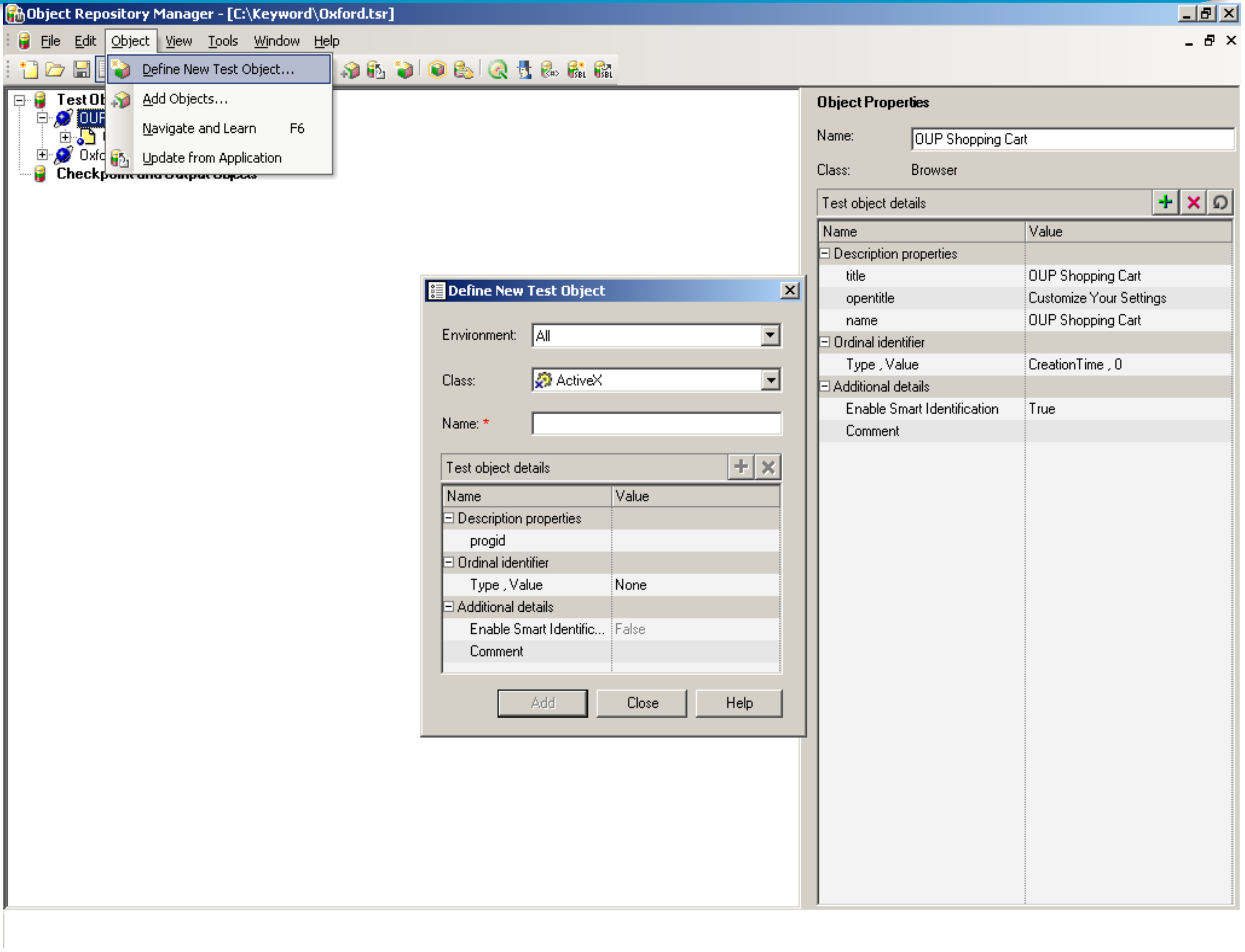
Item	Operation	Value	Documentation
Action1			
Flight Reservation			
MaskEdBox	Type		Type "120507" in the "MaskEdBox" ActiveX object.
Fly From:	Select	"Frankfurt"	Select the "Frankfurt" item from the "Fly From:" list.
Fly To:	Select	"London"	Select the "London" item from the "Fly To:" list.
FLIGHT	Click		Click the "FLIGHT" button.
Flights Table			
Name:	Set	"nonkjk"	Enter "nonkjk" in the "Name:" edit box.
Insert Order	Click		Click the "Insert Order" button.

Logical Name of the object

Flight Reservation — Flight Schedule: Date of Flight: 12/05/07

Object Repository Manager - Define New Test Object

Define New Test Objects helps in adding Test object that do not yet exist in application. This enables to prepare an object repository and build tests or components for application before the application is ready for testing. Objects can be added to Local or Shared Object Repository


Object Repository Manager - [C:\Keyword\Oxford.tsr]
File Edit Object View Tools Window Help
Define New Test Object...
Add Objects...
Navigate and Learn F6
Update from Application
Test Ob
OUP
Oxfo
Checkpoint and Output Objects
Add
Close
Help

Object Properties

Name:	OUP Shopping Cart
Class:	Browser

Test object details

Name	Value
Description properties	
title	OUP Shopping Cart
opentitle	Customize Your Settings
name	OUP Shopping Cart
Ordinal identifier	
Type , Value	CreationTime , 0
Additional details	
Enable Smart Identification	True
Comment	

Define New Test Object

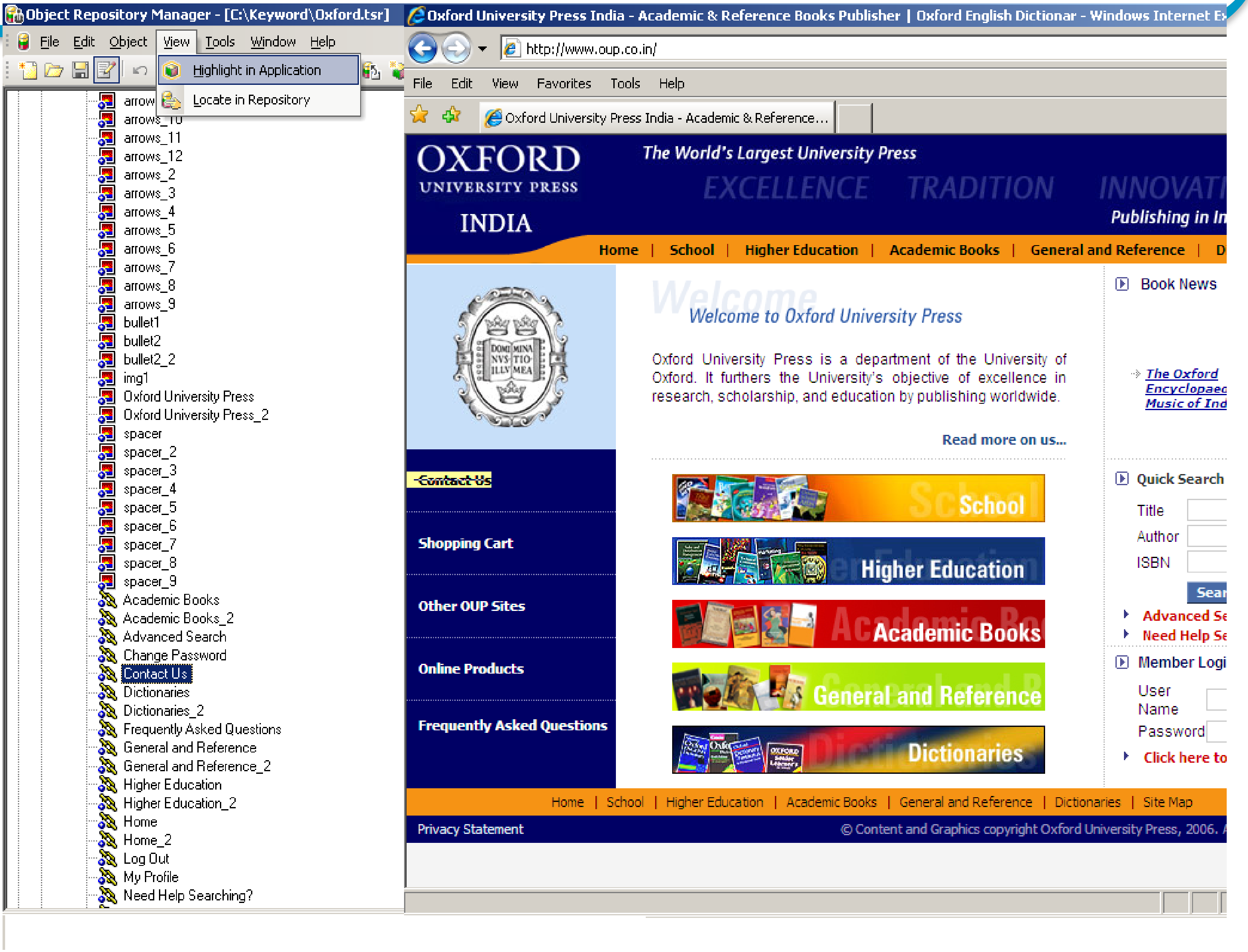
Environment:	All
Class:	ActiveX
Name: *	

Test object details

Name	Value
Description properties	
progid	
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identific...	False
Comment	

Object Repository Manager - Highlight in Application

Select a test object in object repository and highlight it in the application. When highlight in Application is selected, QTP indicates the selected object's location in the application by temporarily showing a frame around the object and causing it to flash briefly. The application must be open to the correct context so that the object is visible.

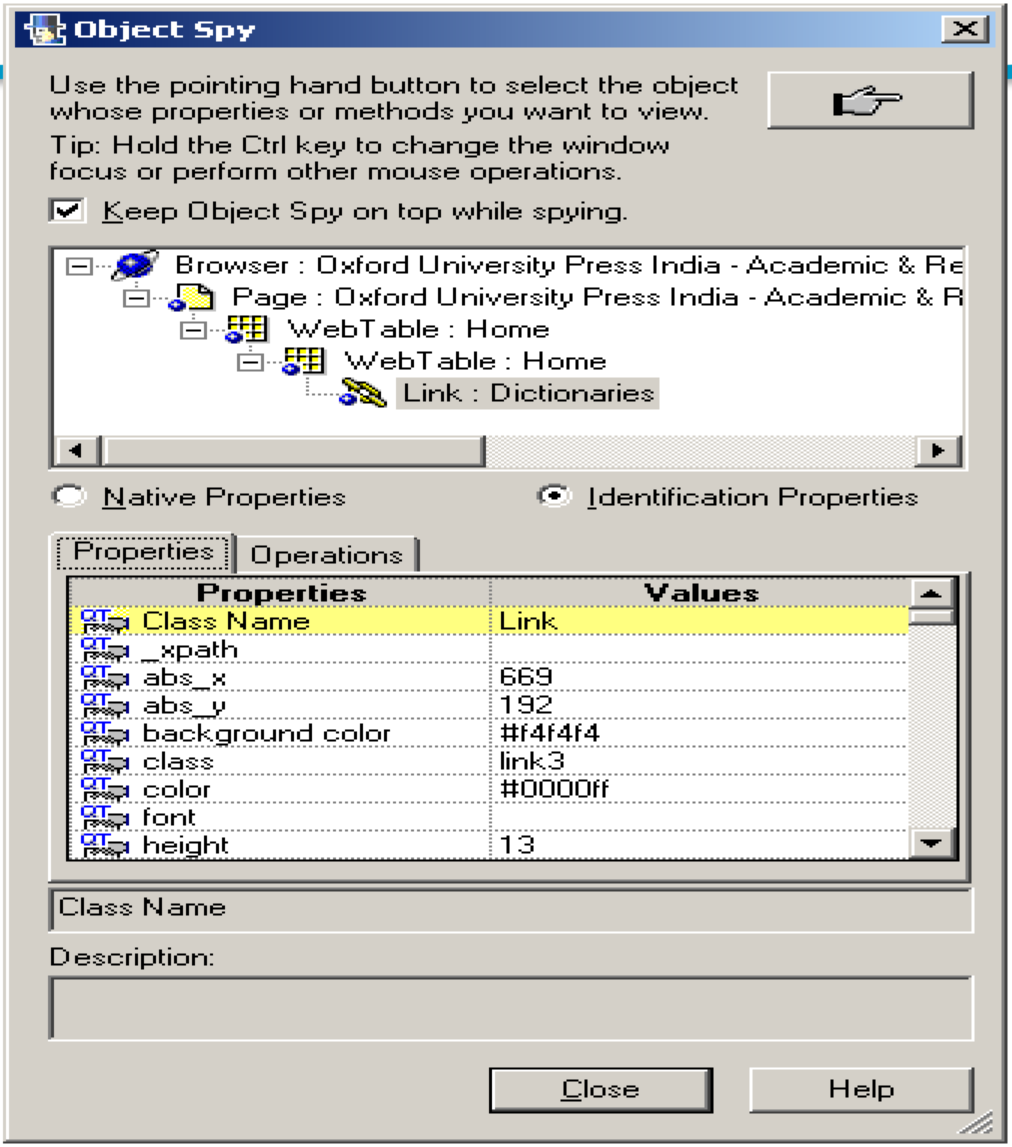


Oxford University Press is a department of the University of Oxford. It furthers the University's objective of excellence in research, scholarship, and education by publishing worldwide.

Object Repository Manager - Object Spy

It enables 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 QTP uses to represent that object.

This helps in finding the current object properties of any test object.



Use the pointing hand button to select the object whose properties or methods you want to view.

Tip: Hold the Ctrl key to change the window focus or perform other mouse operations.

[✓] Keep Object Spy on top while spying.

```
Browser : Oxford University Press India - Academic & Re
  Page : Oxford University Press India - Academic & R
    WebTable : Home
      WebTable : Home
        Link : Dictionaries
```

Native Properties [✓ Identification Properties]

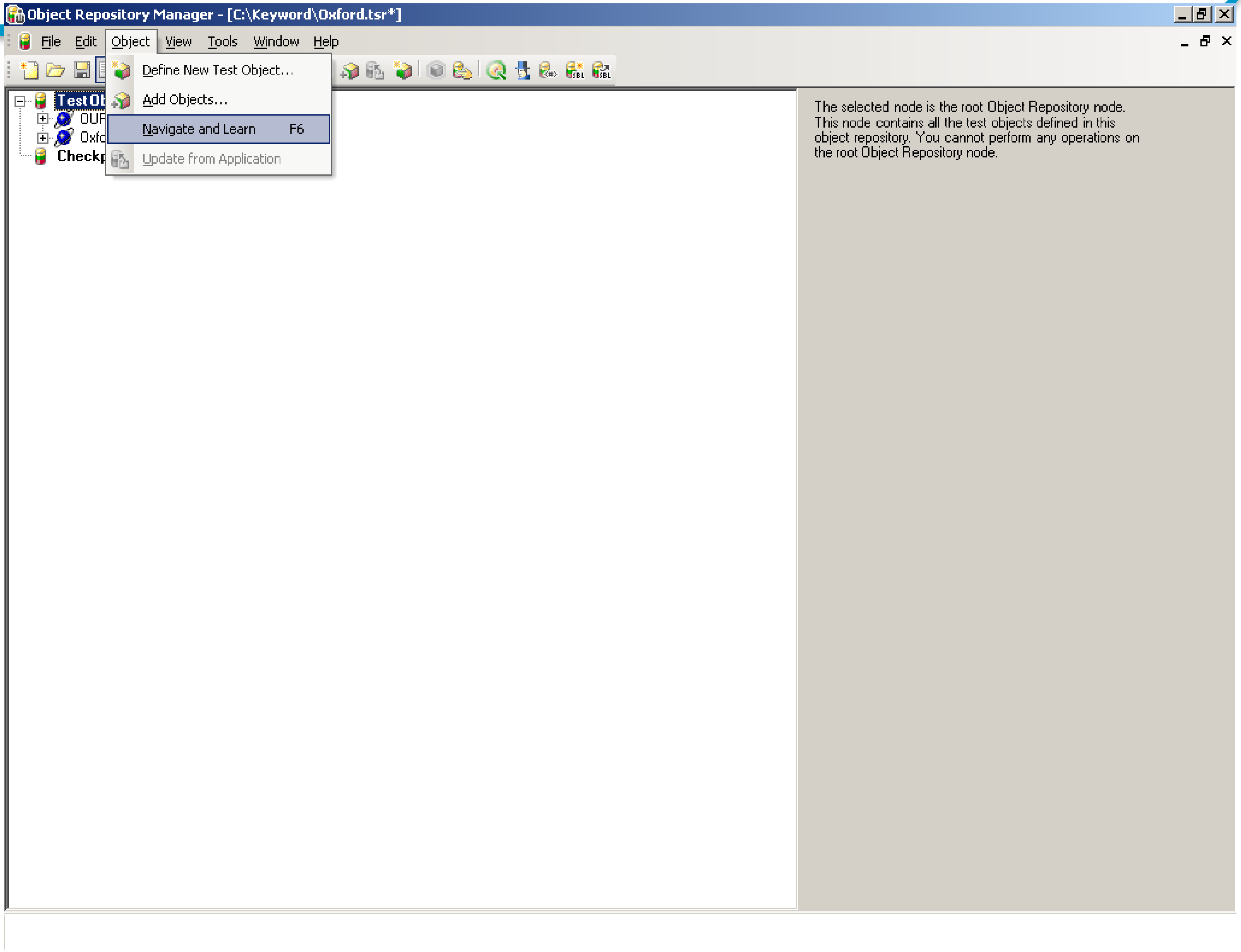
Properties	Values
Class Name	Link
_xpath	
abs_x	669
abs_y	192
background color	#f4f4f4
class	link3
color	#0000ff
font	
height	13

Class Name

Description:

Object Repository Manager - Navigate and Learn

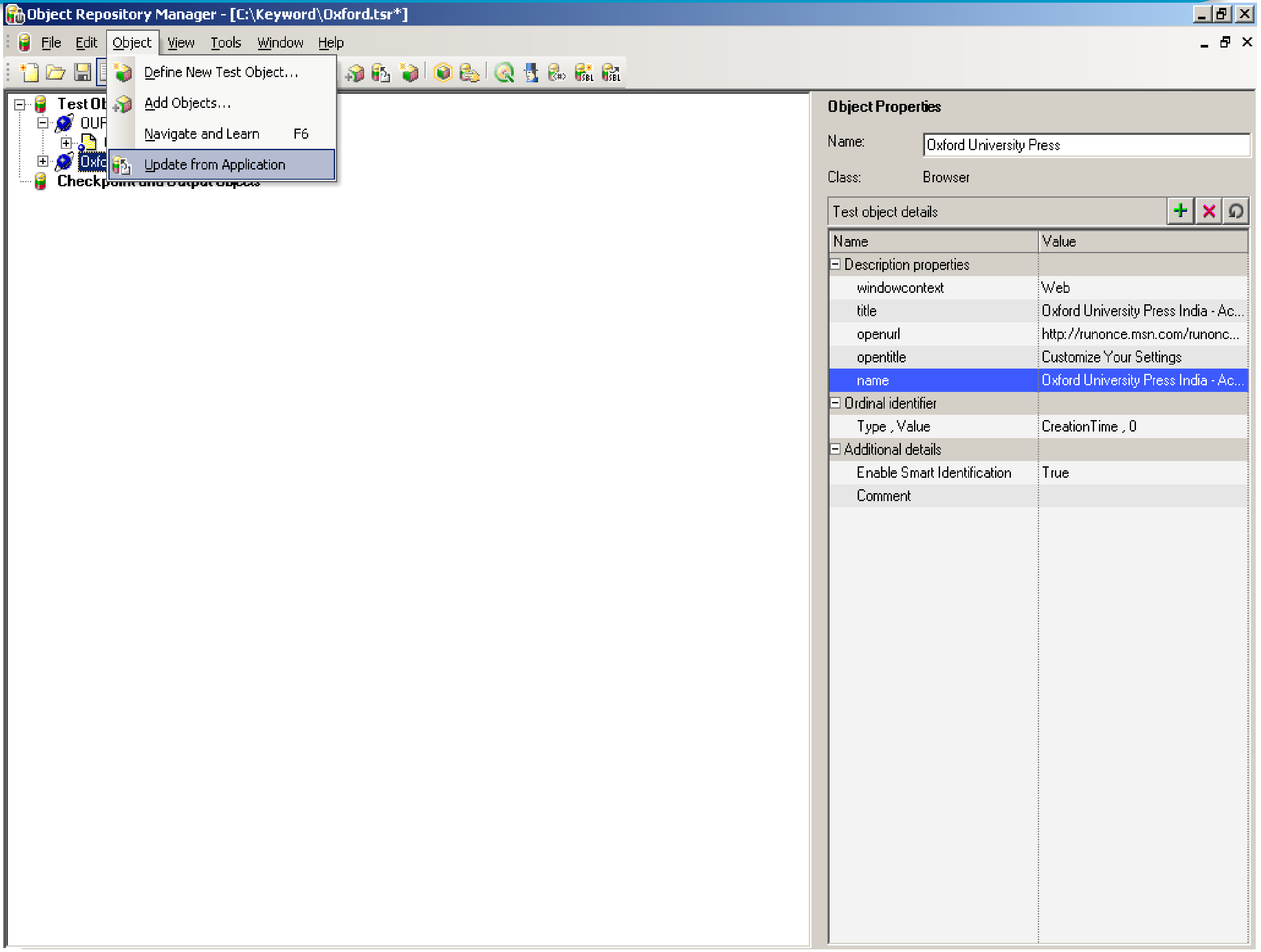
The Navigate and Learn option enables to add multiple test objects to a shared object repository based on defined filter while navigating through the application.


Object Repository Manager - [C:\Keyword\Oxford.tsr*]
File Edit Object View Tools Window Help
Define New Test Object...
Add Objects...
Navigate and Learn F6
Update from Application
Test Ob
OUF
Oxfo
Checkp

The selected node is the root Object Repository node. This node contains all the test objects defined in this object repository. You cannot perform any operations on the root Object Repository node.

Object Repository Manager - Update from Application

As the application changes, it is important to update individual test object properties from the object in the application using the Update from Application option


Object Repository Manager - [C:\Keyword\Oxford.tsr*]
File Edit Object View Tools Window Help
Define New Test Object...
Add Objects...
Navigate and Learn F6
Update from Application
Test Ob
OUP
Oxfo
Checkpoint and Output Objects
Object Properties
Test object details

Name: Oxford University Press

Class: Browser

Name	Value
Description properties	
windowcontext	Web
title	Oxford University Press India - Ac...
openurl	http://runonce.msn.com/runonc...
opentitle	Customize Your Settings
name	Oxford University Press India - Ac...
Ordinal identifier	
Type , Value	CreationTime , 0
Additional details	
Enable Smart Identification	True
Comment	

Introduction to Synchronization

When we run test it might happen that the time taken by the application to respond is more and the images might not be loaded in which case QTP will report an error as it will not be able to find the object during run time. In such cases we need to synchronize our tests.

We can synchronize using the following options –

1. Insert a synchronization point
2. Use EXIST or WAIT statements
3. Modify the default amount of time that Quick Test waits for a Web page to load i.e. Modify object synchronization timeout value

Synchronization can be done for following tasks

•For a progress bar to reach 100%
•For a status message to be displayed
•For a property change of an object
•For a window or pop-up message to be displayed

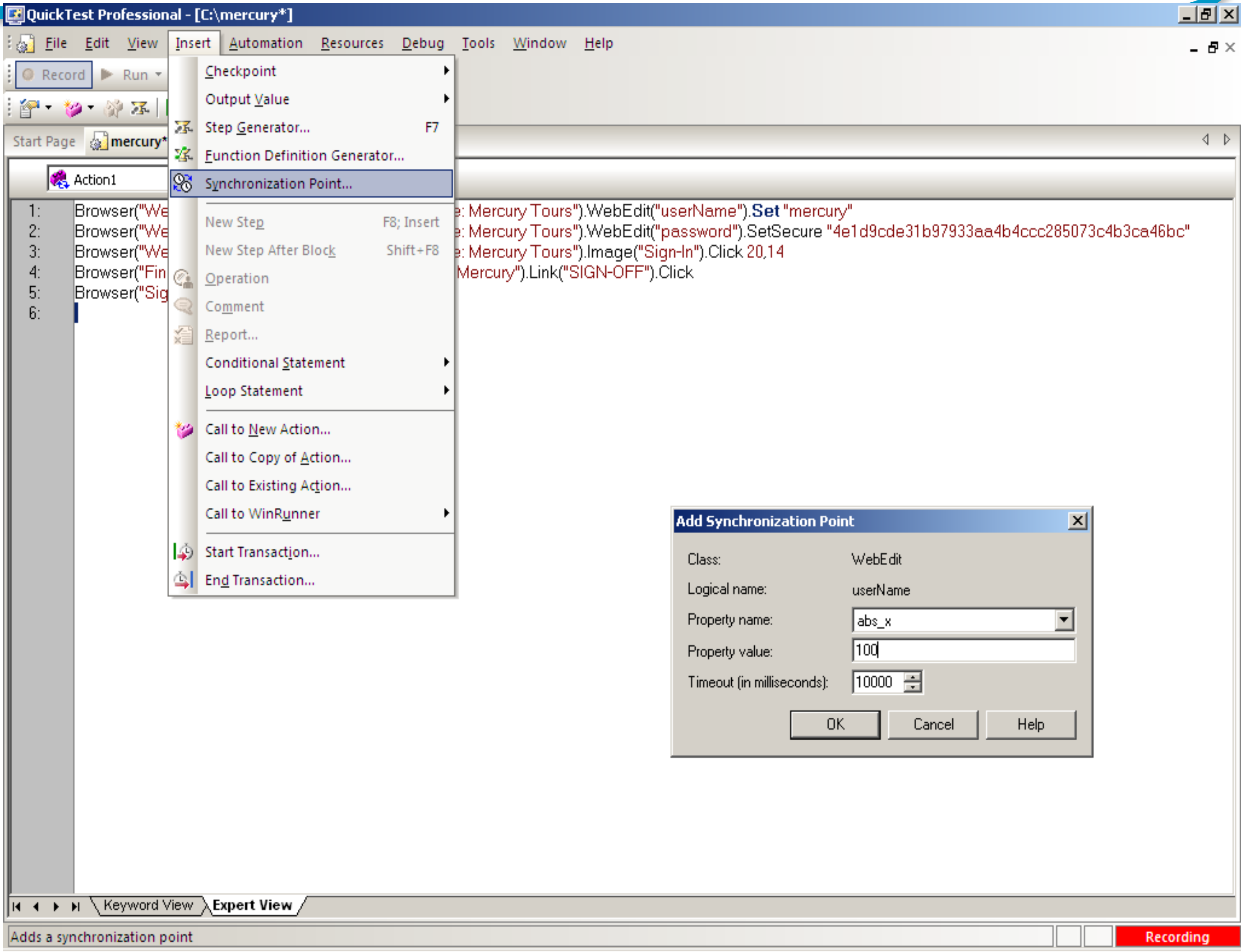
Adding Synchronization Point in Recording mode

Insert while recording by selecting Insert → Synchronization point

Click on the control to be synchronized.

Add Synchronization Point dialog is displayed.

Code snippet Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").WaitProperty "abs_x", 100, 10000



```
Browser("We                    : Mercury Tours").WebEdit("userName").Set "mercury"
Browser("We                    : Mercury Tours").WebEdit("password").SetSecure "4e1d9cde31b97933aa4b4ccc285073c4b3ca46bc"
Browser("We                    : Mercury Tours").Image("Sign-In").Click 20,14
Browser("Fin                    Mercury").Link("SIGN-OFF").Click
Browser("Sig
```

Add Synchronization Point

Class:	WebEdit
Logical name:	userName
Property name:	abs_x
Property value:	100
Timeout (in milliseconds):	10000

Synchronization in Expert View

Synchronization – Using Wait

The timing problem can be handled by adding a Wait statement in the script instead of inserting a synchronization point.

Consider the same script, a wait statement is included to instruct the tool to wait for 2 seconds.

Example

```
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").Image("Sign-In").Click

Wait (2)

Browser("Find a Flight: Mercury").Page("Find a Flight: Mercury").Link("SIGN-OFF").Click
```

Synchronization in Expert View (contd)

Synchronization - Using Exist

Using this function we can check for the existence of an object or a window and continue with the script based on the result.

Syntax: Object.Exist(Timeout in seconds)

Object can be any GUI object or Window

The function returns a Boolean value. True value is returned in case object exists else False is returned.

Time Out – time for which the object's existence should be checked

Example:

```
If Window("Flight Reservation").WinButton("Update Order").Exist(10) Then

        ……..

End if
```

Transactions

A transaction represents the process in your application that we are interested in measuring. By defining a transaction we can measure how long it takes to run a section of a test script.

Need for Transactions:
Transactions can be used to measure the performance of the script
By analyzing the output of the Transaction we can optimize the script in certain areas

Defining a transaction:
You define transactions within your test by enclosing the appropriate sections of the test with start and end transaction statements. During the test run, the StartTransaction step signals the beginning of the time measurement.
The time measurement continues until the EndTransaction step is reached.

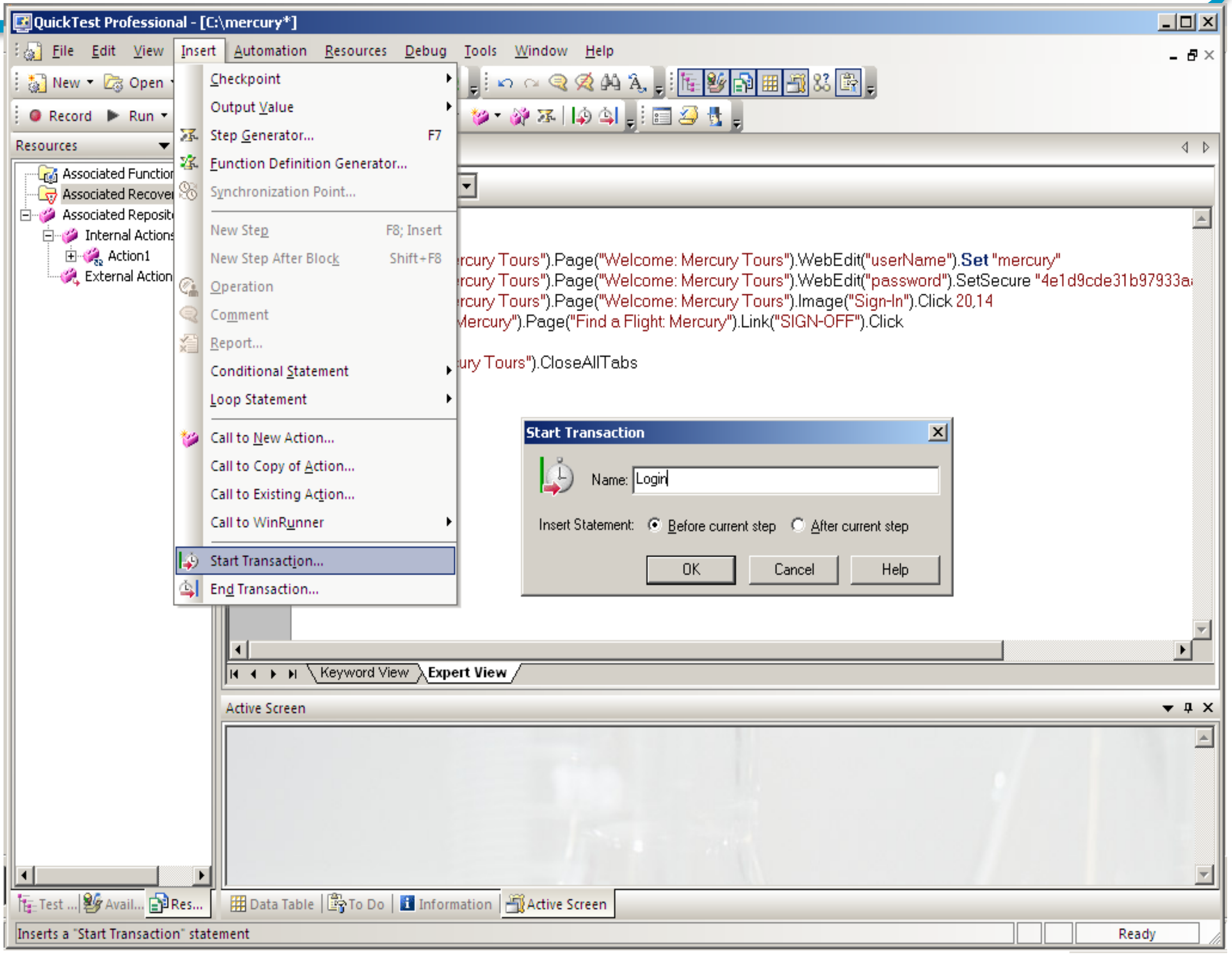
Creating Transaction

Select Insert->Start Transaction menu.

Start Transaction dialog is displayed. Enter valid name for the transaction.

Click OK button

Code snippet: Services.StartTransaction "Login"



```
rcury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").Set "mercury"
rcury Tours").Page("Welcome: Mercury Tours").WebEdit("password").SetSecure "4e1d9cde31b97933a
rcury Tours").Page("Welcome: Mercury Tours").Image("Sign-In").Click 20,14
Mercury").Page("Find a Flight: Mercury").Link("SIGN-OFF").Click

ury Tours").CloseAllTabs
```

Start Transaction

Name: Login

Insert Statement: [✓ Before current step] After current step

OK Cancel Help

Inserts a "Start Transaction" statement

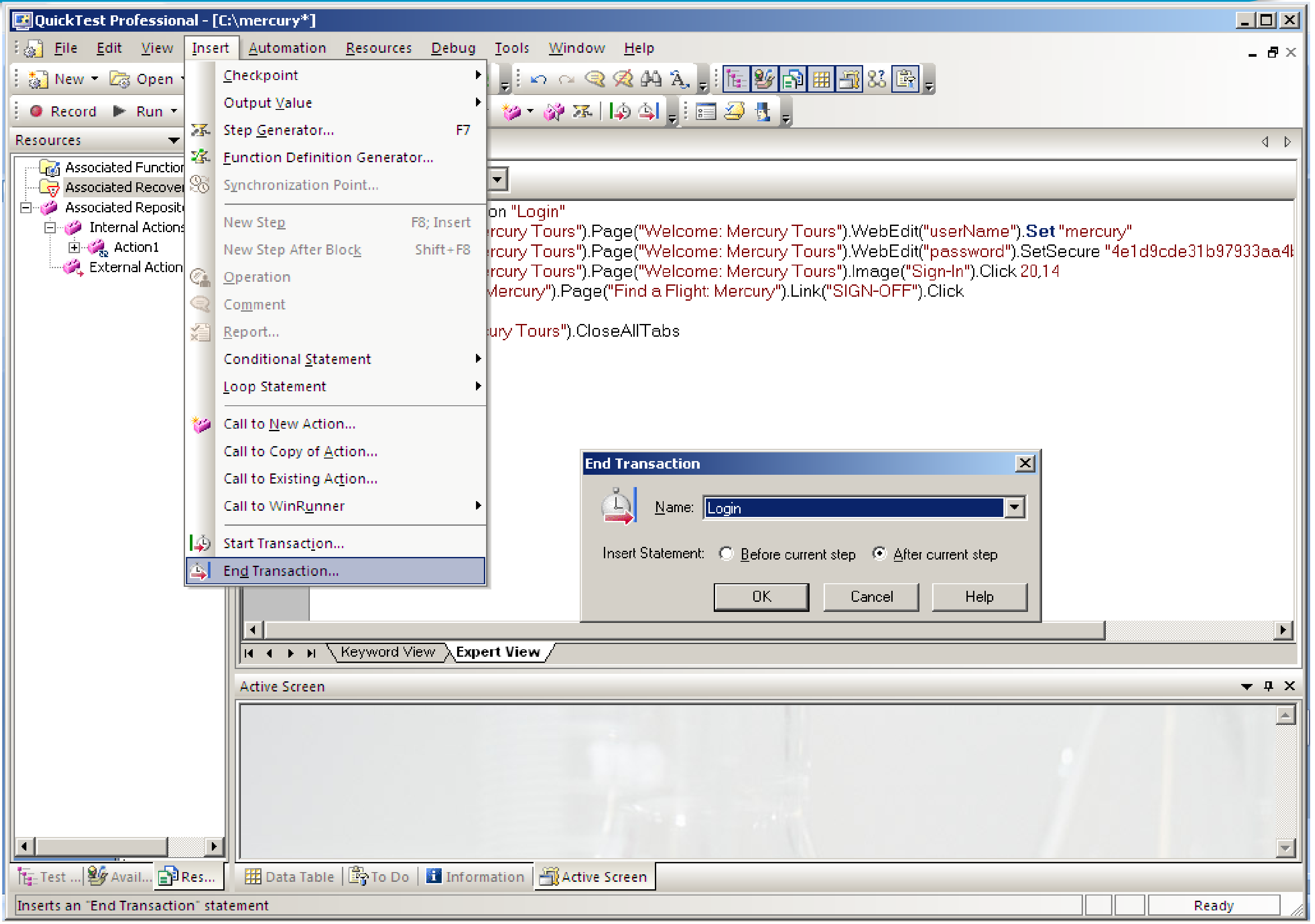
End Transaction

Select Insert->End Transaction menu.

End Transaction dialog is displayed. Select the transaction.

Click OK button

Code snippet: Services.EndTransaction "Login"



```
on "Login"
rcury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").Set "mercury"
rcury Tours").Page("Welcome: Mercury Tours").WebEdit("password").SetSecure "4e1d9cde31b97933aa4b
rcury Tours").Page("Welcome: Mercury Tours").Image("Sign-In").Click 20,14
Mercury").Page("Find a Flight: Mercury").Link("SIGN-OFF").Click

ury Tours").CloseAllTabs
```

End Transaction

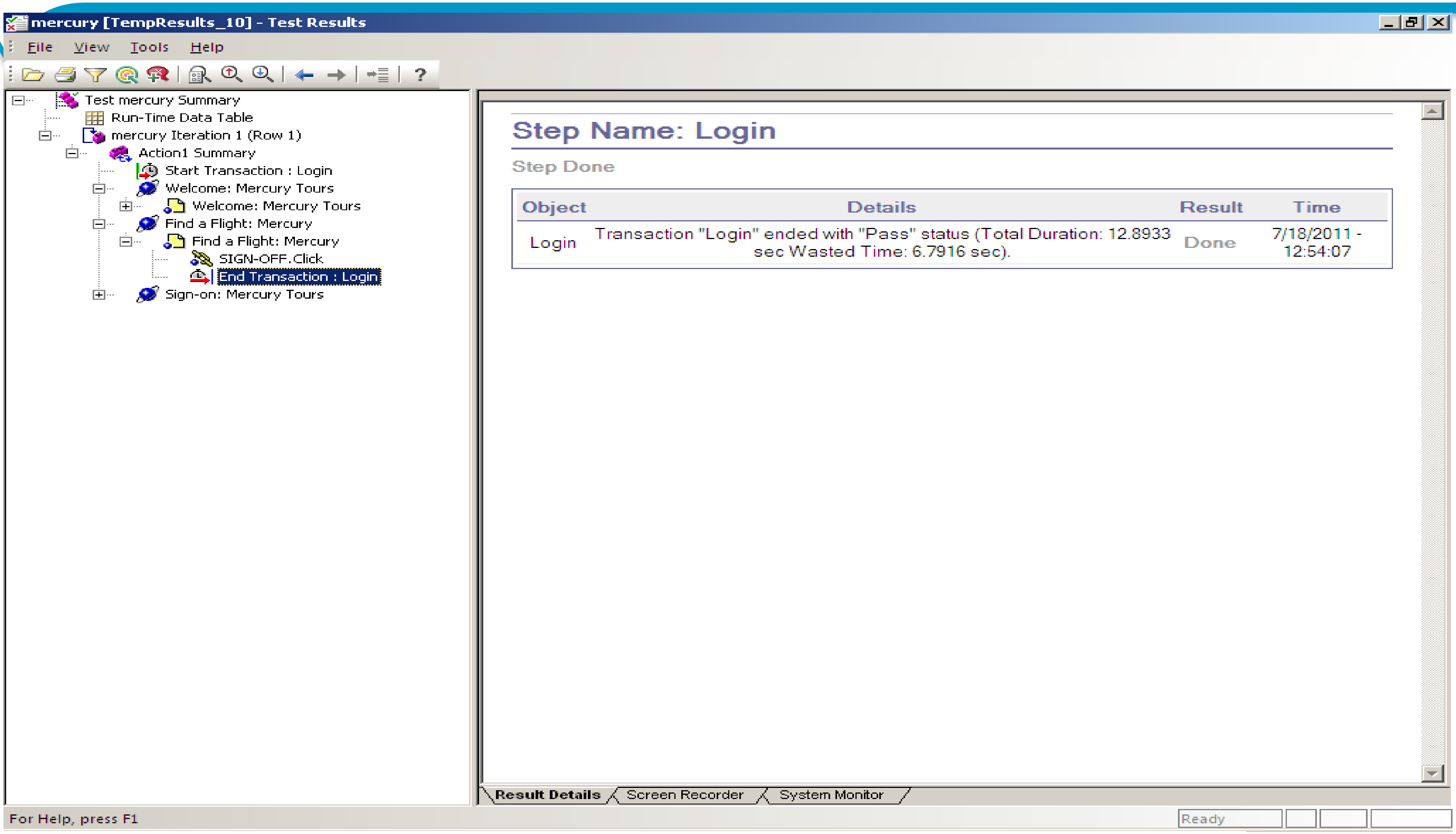
Name: Login

Insert Statement: Before current step [✓ After current step]

OK Cancel Help

Inserts an "End Transaction" statement

Transaction execution in Result pane



mercury [TempResults_10] - Test Results

```
Test mercury Summary
    Run-Time Data Table
    mercury Iteration 1 (Row 1)
        Action1 Summary
            Start Transaction : Login
            Welcome: Mercury Tours
                Welcome: Mercury Tours
            Find a Flight: Mercury
                Find a Flight: Mercury
                    SIGN-OFF.Click
                    End Transaction : Login
            Sign-on: Mercury Tours
```

Step Name: Login

Step Done

Object	Details	Result	Time
Login	Transaction "Login" ended with "Pass" status (Total Duration: 12.8933 sec Wasted Time: 6.7916 sec).	Done	7/18/2011 - 12:54:07

Introduction to Checkpoints

It is a step in QTP that compares two values and then results are reported.

A checkpoint in QTP is a verification point that will compare the current value for a property with the expected value for that property.

It also compares the actual and expected results and if the two values passes, the checkpoint pass else if mismatch checkpoint fails.

Types of Checkpoint

Check Point Type	Description	Example
Standard Checkpoint	Checks property values of an object's properties	Check that a radio button is selected.
Table Checkpoint	Checks information in a table	Check that the value in a table cell is correct.
Page checkpoint	Checks the characteristics of a Web page	Check how long a Web page takes to load or if a Web page contains broken links.
Text / Text Area Checkpoint	Checks that a text string is displayed in the appropriate place in a Web page or application window	Check whether the expected text string is displayed in the expected location on a Web page or dialog box.
Accessibility Checkpoint	Checks compliance with World Wide Web Consortium (W3C) instructions and guidelines for Web-based technology and information systems	Check if the images on the web page include ALT properties required by the W3C Web content Accessibility Guidelines.

Types of Checkpoint (continued)

Check Point Type	Description	Example
Bitmap Checkpoint	Checks the bitmap of an image or a full web page. It does a pixel by pixel comparison between actual and expected bitmaps.	Check that a Web page (or any portion of it) is displayed as expected.
Database Checkpoint	Checks the contents of databases accessed by an application or Web site	Check that the value in a database query is correct.
XML Checkpoint	Checks the data content of XML documents	XML file checkpoints are used to check a specified XML file; XML application checkpoints are used to check an XML document within a Web page.

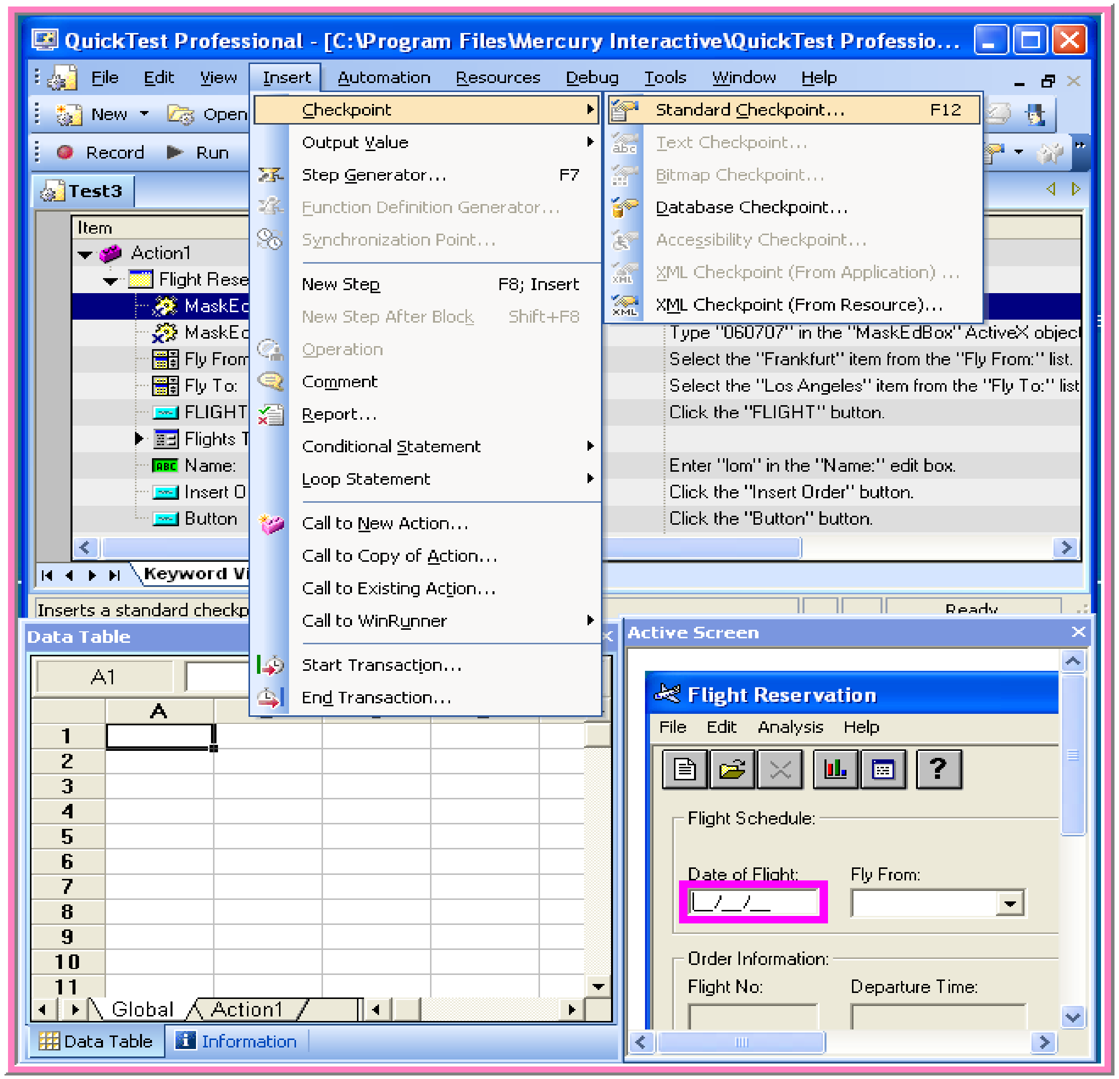
Inserting a Checkpoint

The checkpoint can be inserted by the followings ways:

Select Insert Menu → Select Checkpoint and select the type of checkpoint that needs to be added.

Or

In the Keyword view, Right click on the step where the checkpoint is required and select Insert standard checkpoint

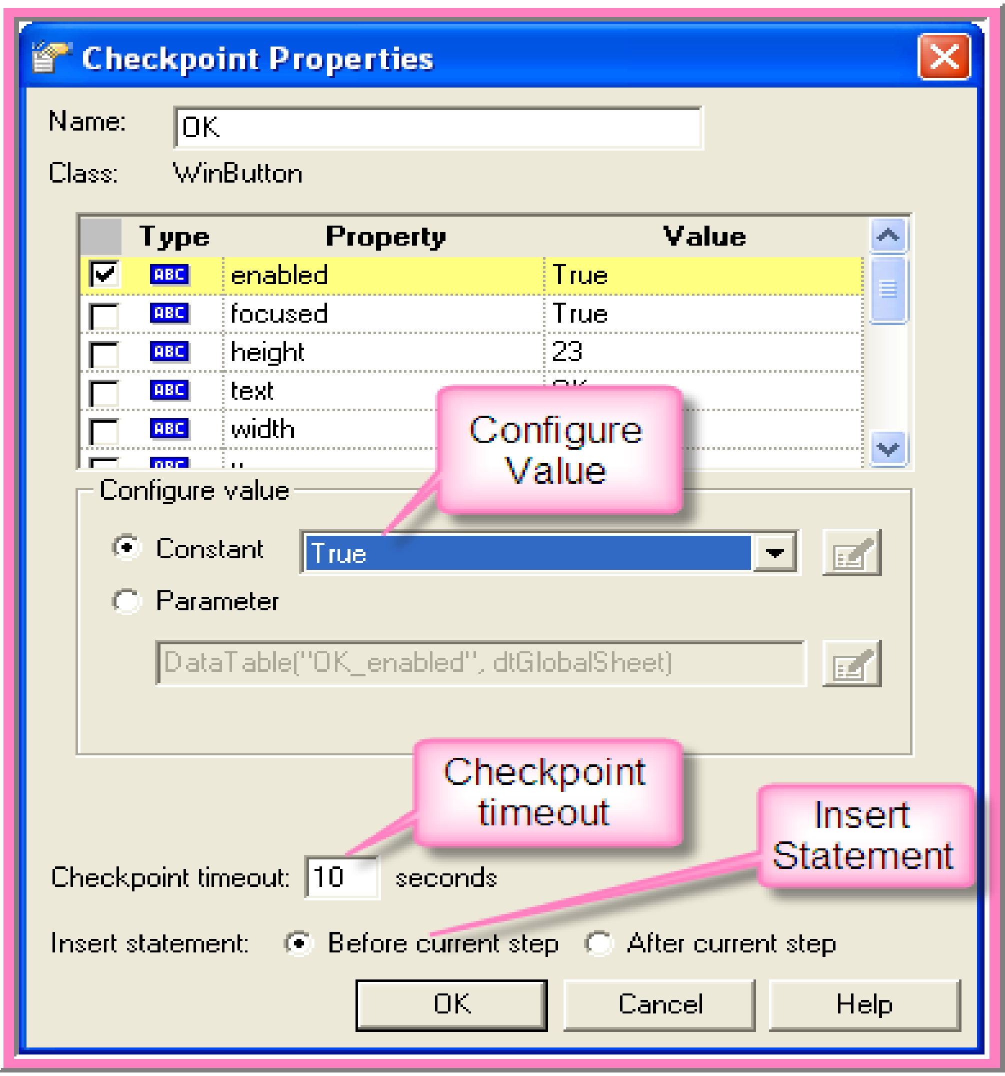


Type "060707" in the "MaskEdBox" ActiveX objec
Select the "Frankfurt" item from the "Fly From:" list.
Select the "Los Angeles" item from the "Fly To:" list
Click the "FLIGHT" button.
Enter "lom" in the "Name:" edit box.
Click the "Insert Order" button.
Click the "Button" button.

Inserts a standard checkp

Checkpoint properties dialog

- **Configure Value→** The value for each property can be configured. The values can be constant or parameterized.
- **Checkpoint timeout →** specifies the time interval during which QTP attempts to perform the checkpoint successfully.
- **Insert statement →** specifies when to perform the checkpoint in the test or component.



Name: OK
Class: WinButton

	Type	Property	Value
✓	ABC	enabled	True
	ABC	focused	True
	ABC	height	23
	ABC	text	OK
	ABC	width	

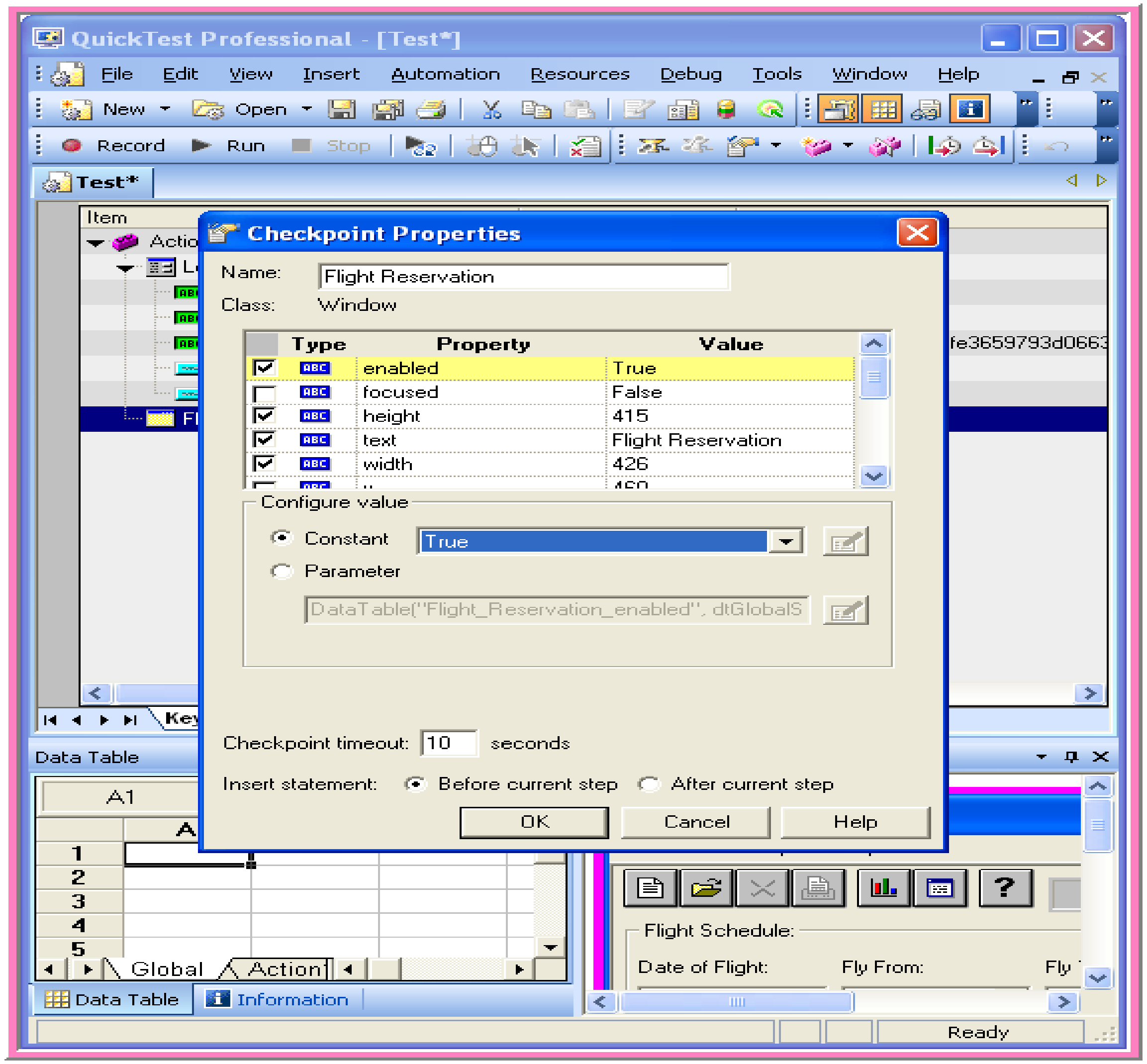
Configure value
[✓ Constant] True
Parameter DataTable("OK_enabled", dtGlobalSheet)

Checkpoint timeout: 10 seconds

Insert statement: [✓ Before current step] After current step

Addition of properties on Checkpoint Properties dialog

1. Select Insert → Standard Checkpoint menu.
2. The checkpoint properties dialog box opens
3. If a Checkpoint needs to be placed for a property. Check that property in the Checkpoint Property window.
4. Modify the expected values as required and Click Ok button to add the checkpoint in the script.


QuickTest Professional - [Test*]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Test*
Item
Data Table
Global Action1
Data Table Information
Flight Schedule:
Date of Flight: Fly From:
Ready

Checkpoint Properties

Name: Flight Reservation
Class: Window

	Type	Property	Value
✓	ABC	enabled	True
	ABC	focused	False
✓	ABC	height	415
✓	ABC	text	Flight Reservation
✓	ABC	width	426

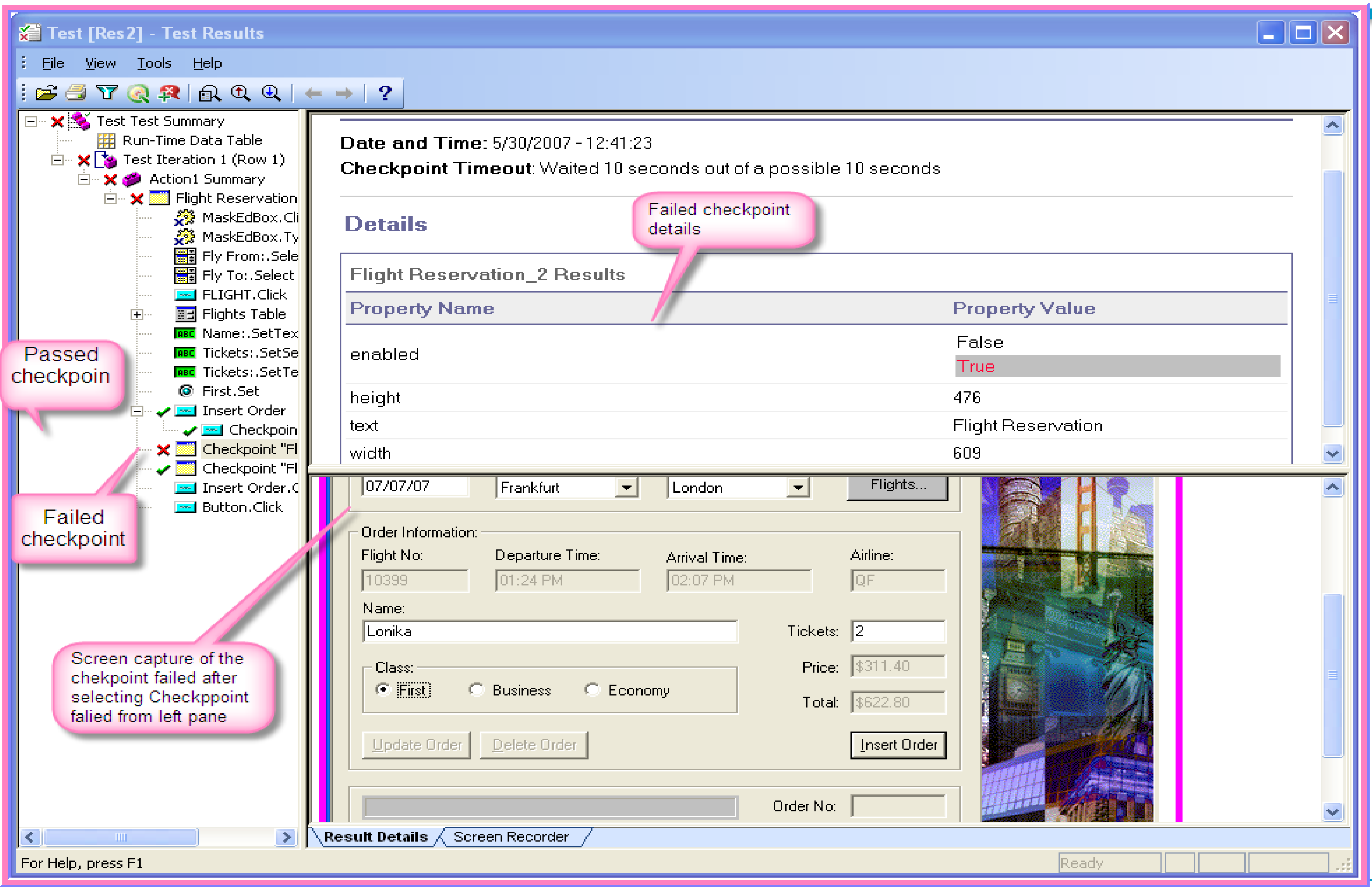
Configure value
[✓ Constant] True
Parameter DataTable("Flight_Reservation_enabled", dtGlobalS

Checkpoint timeout: 10 seconds

Insert statement: [✓ Before current step] After current step

OK Cancel Help

Results of Standard Checkpoint



Date and Time: 5/30/2007 - 12:41:23

Checkpoint Timeout: Waited 10 seconds out of a possible 10 seconds

Details

Flight Reservation_2 Results

Property Name	Property Value
enabled	False True
height	476
text	Flight Reservation
width	609

07/07/07	Frankfurt	London	Flights...

Order Information:

Flight No:	Departure Time:	Arrival Time:	Airline:
10399	01:24 PM	02:07 PM	QF

Name: Lonika

Tickets: 2

Class: [✓ First] Business Economy

Price: $311.40

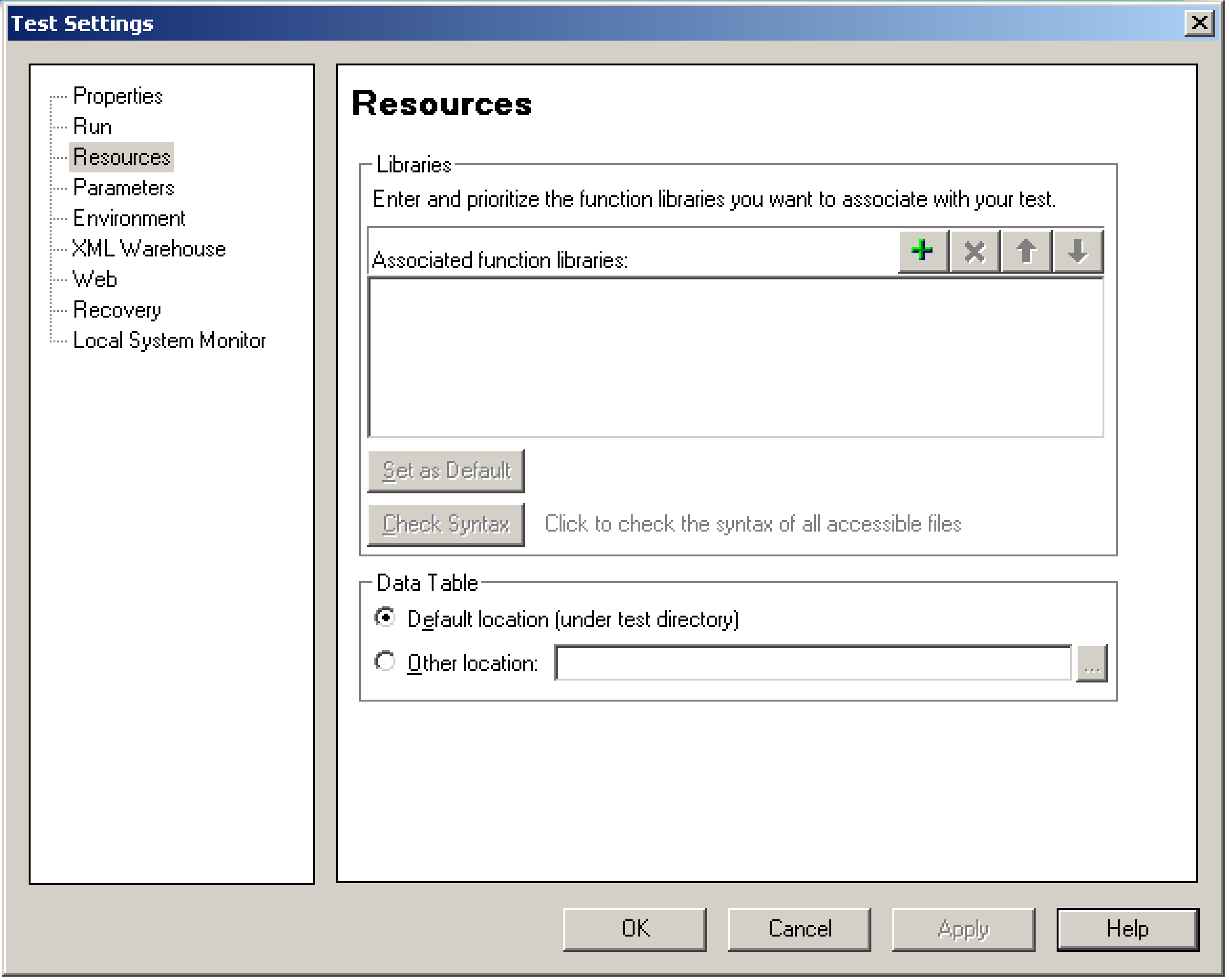
Total: $622.80

Update Order | Delete Order | Insert Order

Order No:

Test Settings – Resources tab

On Resources pane of the Test Settings dialog box we can associate specific files with your test, such as VBScript function libraries and Data Table files. We can also set currently associated function library settings as the default settings for all new tests.
We can check the syntax of the associated libraries.


Test Settings

- Properties
- Run
- Resources
- Parameters
- Environment
- XML Warehouse
- Web
- Recovery
- Local System Monitor

Resources

Libraries

Enter and prioritize the function libraries you want to associate with your test.

Associated function libraries:

Set as Default

Check Syntax — Click to check the syntax of all accessible files

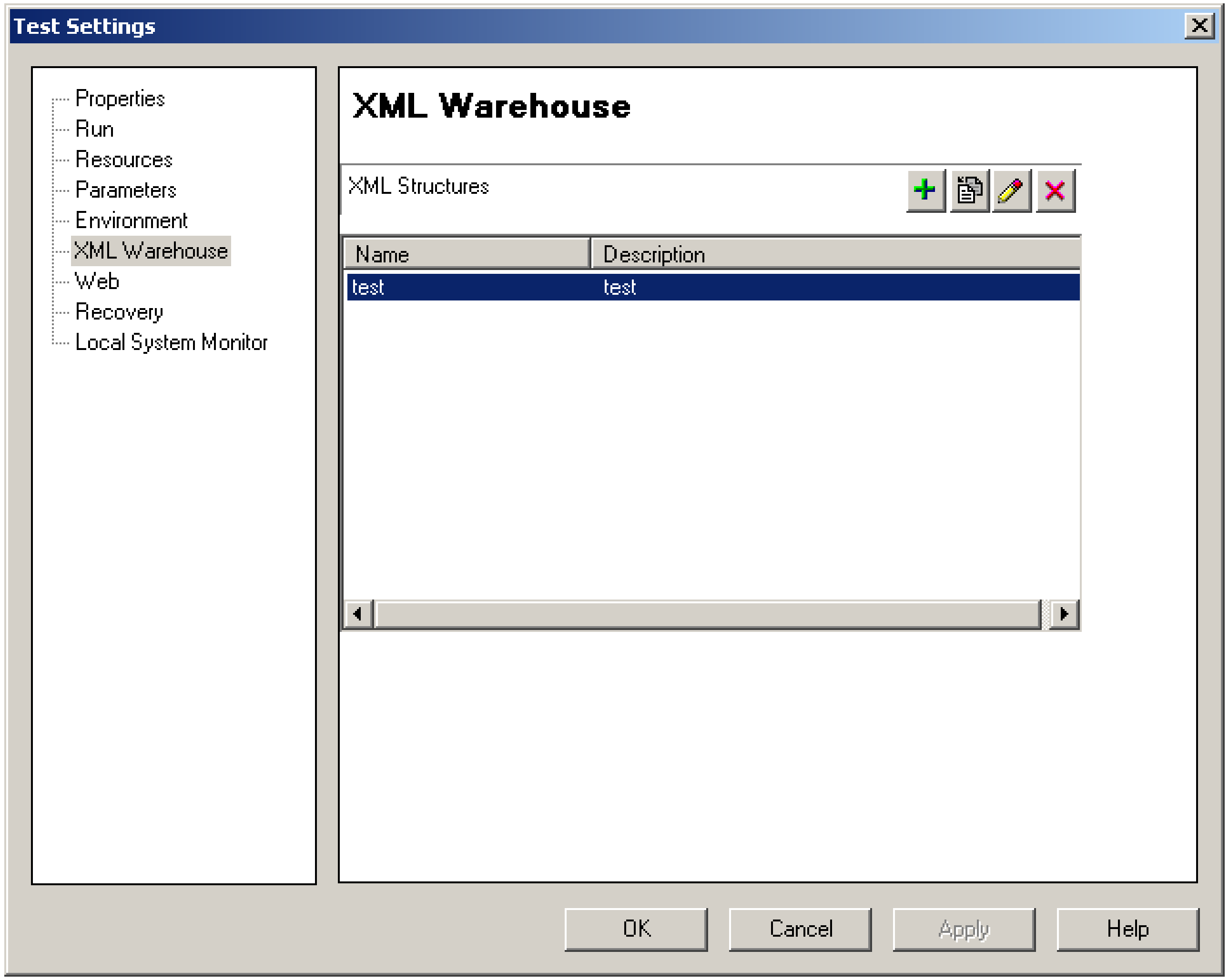
Data Table

[✓ Default location (under test directory)]

Other location:

Test Settings - XML Warehouse tab

We can create XML structures for use as an XML value in our test.
First we need to set up the XML hierarchy for the XML structure by importing it and/or manually adding and editing its nodes.
We can then edit or parameterize the attributes and values of the XML structure.
XML structure can be exported too.



Test Settings

- Properties
- Run
- Resources
- Parameters
- Environment
- XML Warehouse
- Web
- Recovery
- Local System Monitor

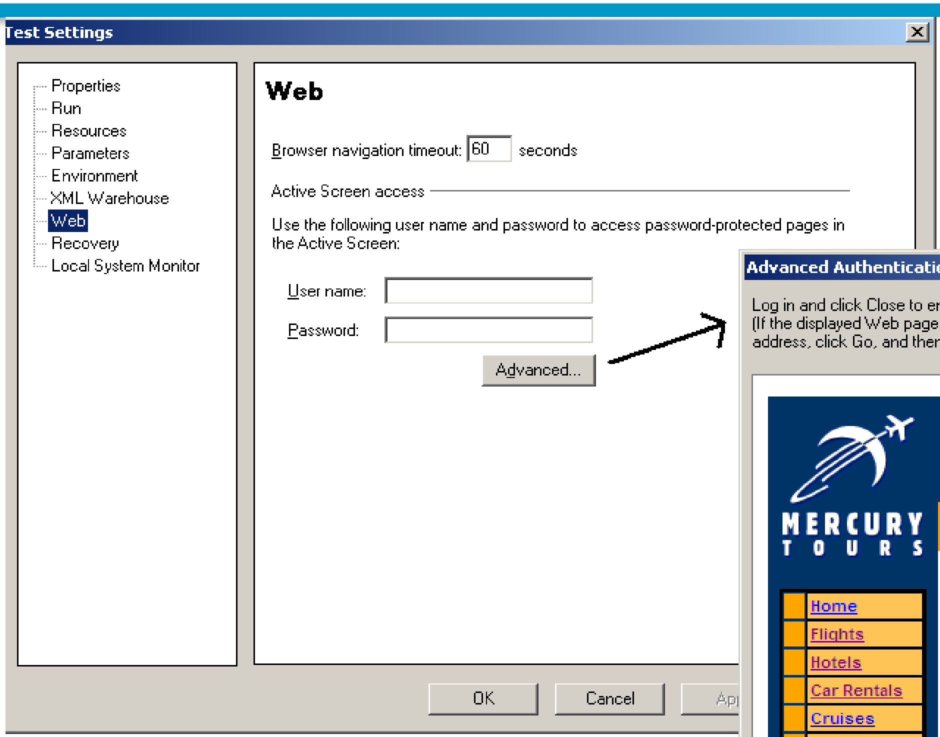
XML Warehouse

XML Structures

Name	Description
test	test

Test Settings – Web tab

On this tab, we can set how long to wait for browser navigations and can also specify the Active Screen access information to use with password-protected resources in the captured Active Screen page.



Test Settings

- Properties
- Run
- Resources
- Parameters
- Environment
- XML Warehouse
- Web
- Recovery
- Local System Monitor

Web

Browser navigation timeout: 60 seconds

Active Screen access

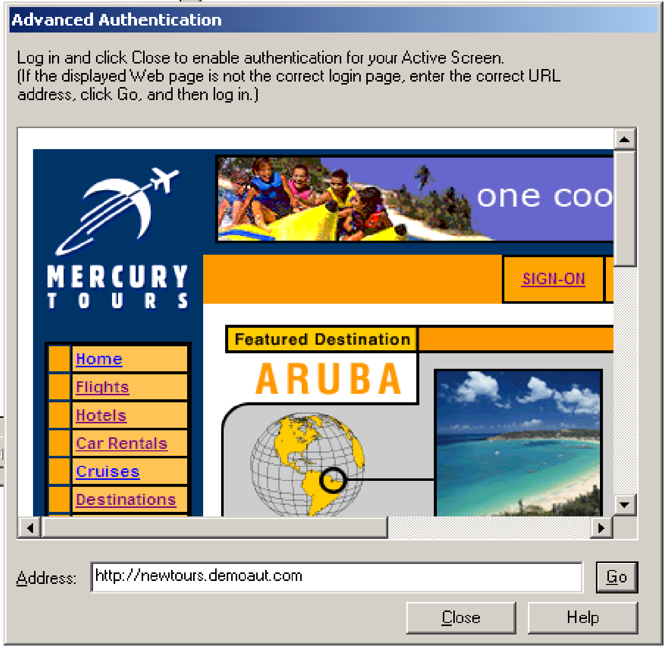
Use the following user name and password to access password-protected pages in the Active Screen:

User name:

Password:

Advanced...

OK Cancel Ap



Advanced Authentication

Log in and click Close to enable authentication for your Active Screen.
(If the displayed Web page is not the correct login page, enter the correct URL address, click Go, and then log in.)

Address: http://newtours.demoaut.com

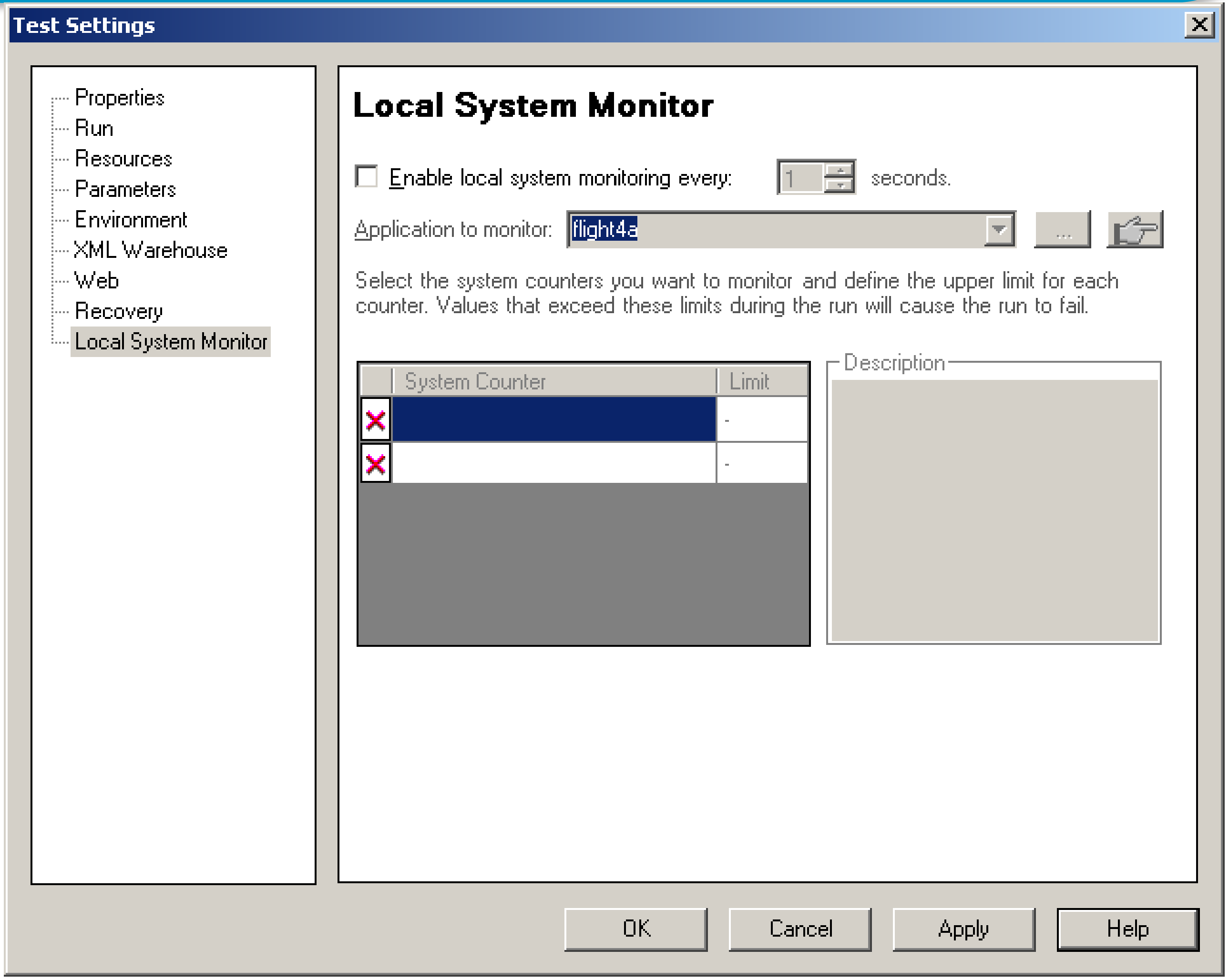
Go

Close Help

Test Settings - Local System Monitor

LSM enables to activate system monitoring and we can define the system counters to be tracked during a run session.

The Local System Monitor data that is captured during a test run is displayed in the Test Results window



Test Settings

- Properties
- Run
- Resources
- Parameters
- Environment
- XML Warehouse
- Web
- Recovery
- Local System Monitor

Local System Monitor

[ ] Enable local system monitoring every: 1 seconds.

Application to monitor: flight4a

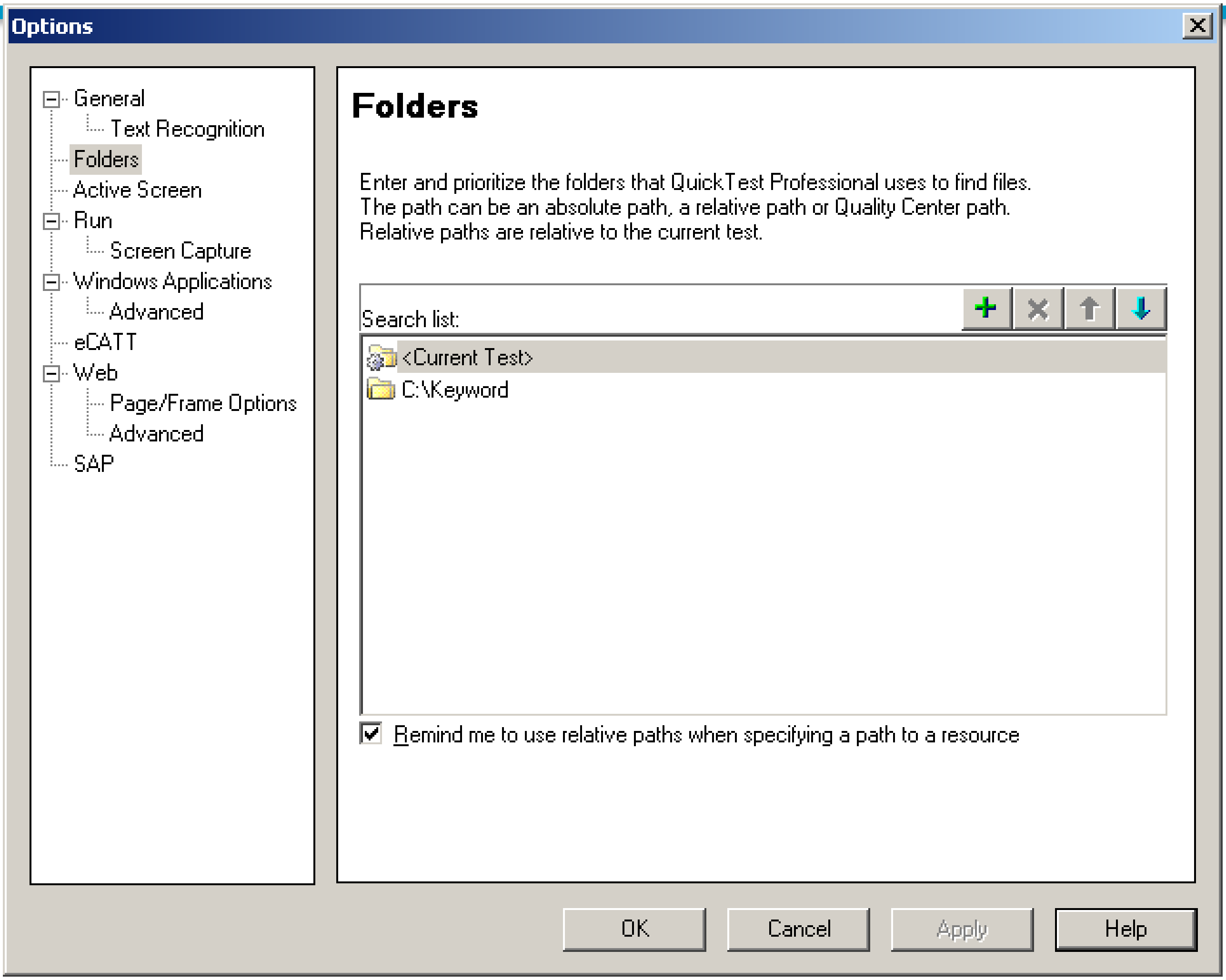
Select the system counters you want to monitor and define the upper limit for each counter. Values that exceed these limits during the run will cause the run to fail.

	System Counter	Limit
✗		-
✗		-

Description

Tools Options – Folder tab

The Folders pane enables to enter the folders (search paths) in which QTP searches for tests, components, actions, or resource files that are specified as relative paths in dialog boxes and steps. If the same file name exists in more than one folder, QTP uses the first instance it finds.



Folders

Enter and prioritize the folders that QuickTest Professional uses to find files.
The path can be an absolute path, a relative path or Quality Center path.
Relative paths are relative to the current test.

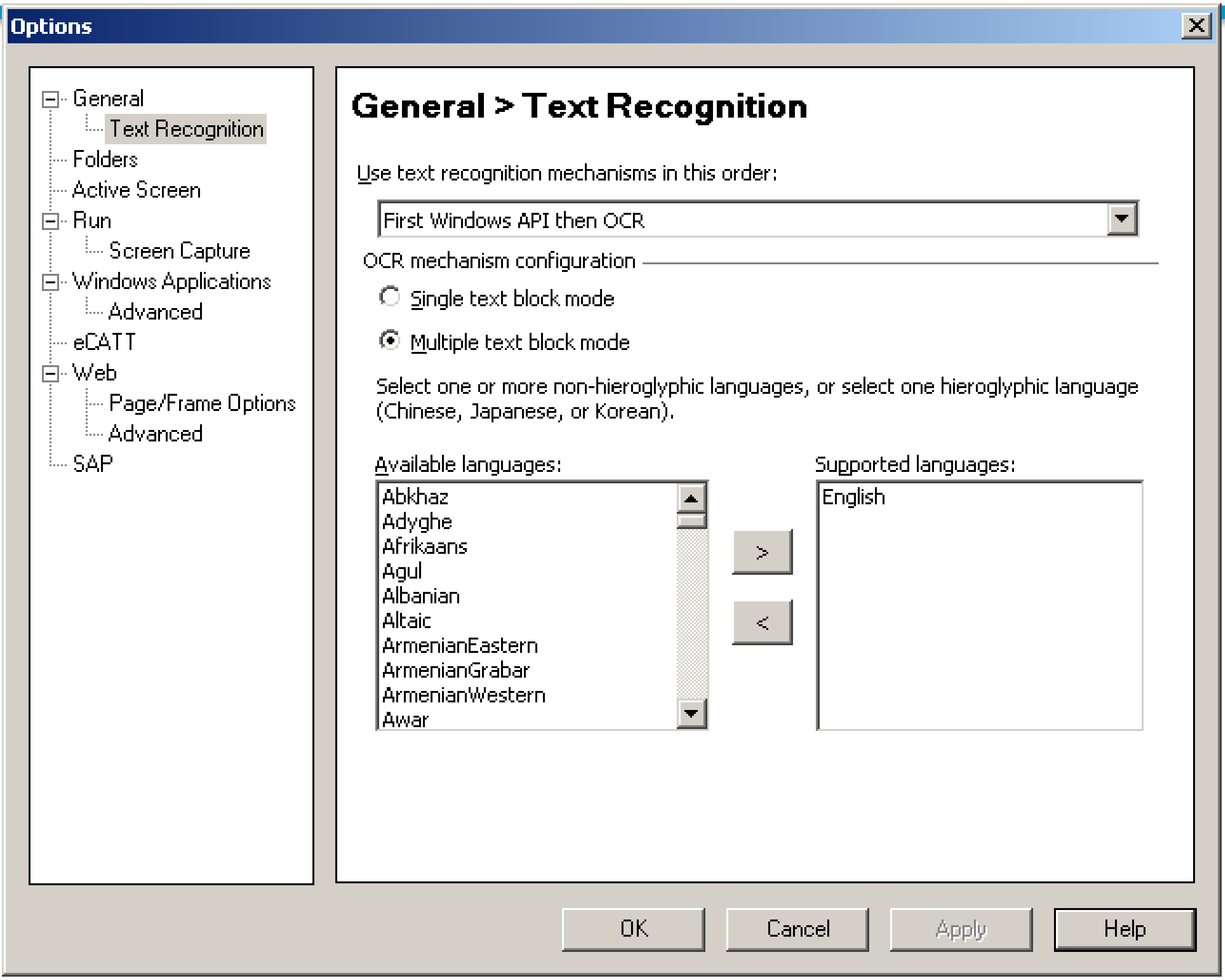
Search list:

<Current Test>
C:\Keyword

☑ Remind me to use relative paths when specifying a path to a resource

Tools Option – General Text Recognition Pane

This pane enables to configure how QuickTest identifies text in your application. This pane can be used to modify the default text capture mechanism, OCR (optical character recognition) mechanism mode, and the language dictionaries the OCR mechanism uses to identify text.



General > Text Recognition

Use text recognition mechanisms in this order: First Windows API then OCR

OCR mechanism configuration

( ) Single text block mode
[✓ Multiple text block mode]

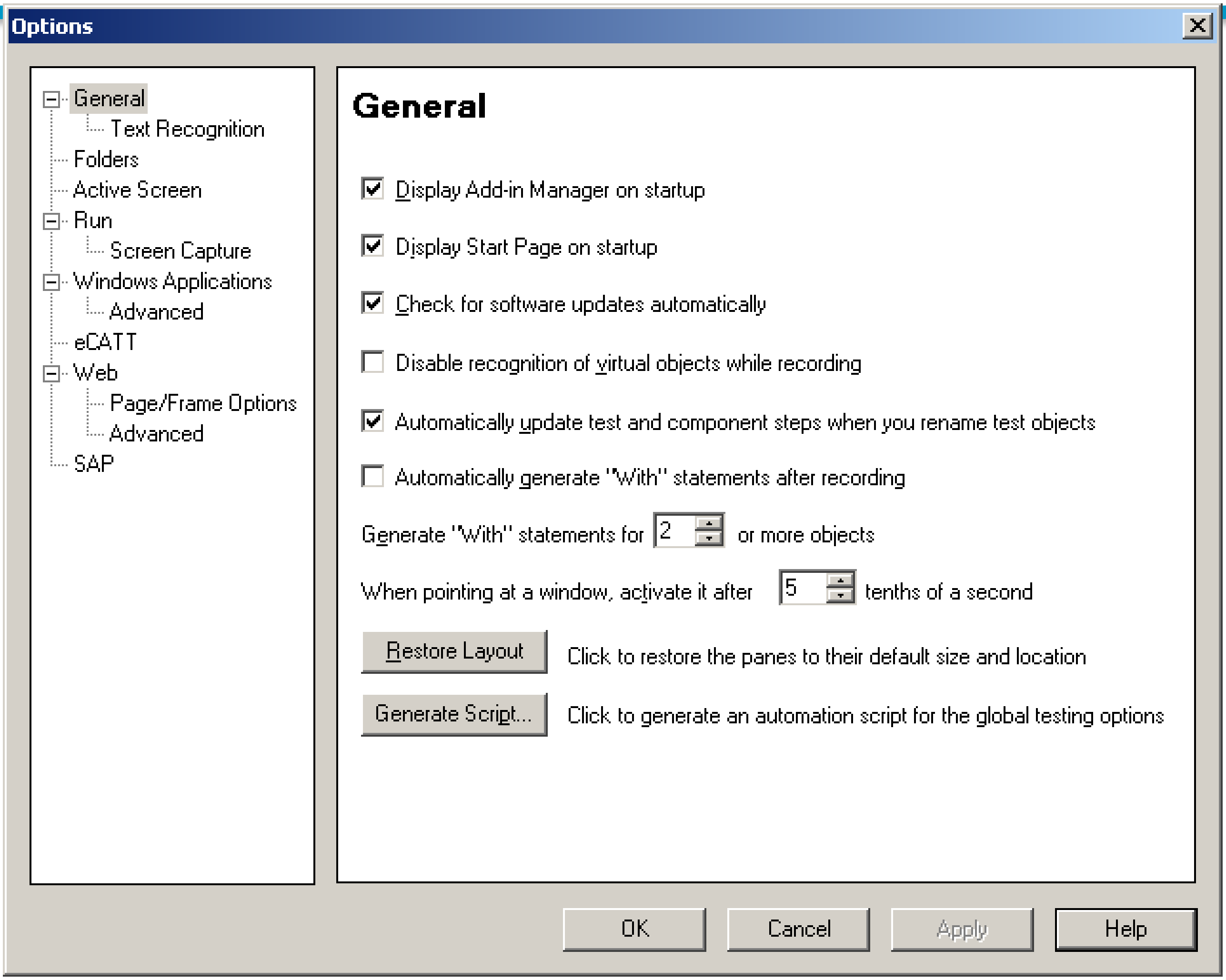
Select one or more non-hieroglyphic languages, or select one hieroglyphic language (Chinese, Japanese, or Korean).

Available languages: Abkhaz, Adyghe, Afrikaans, Agul, Albanian, Altaic, ArmenianEastern, ArmenianGrabar, ArmenianWestern, Awar

Supported languages: English

Tools Option – General tab

The General pane options affect the general appearance of QuickTest and other general testing options.



General

- [✓] Display Add-in Manager on startup
- [✓] Display Start Page on startup
- [✓] Check for software updates automatically
- [ ] Disable recognition of virtual objects while recording
- [✓] Automatically update test and component steps when you rename test objects
- [ ] Automatically generate "With" statements after recording

Generate "With" statements for 2 or more objects

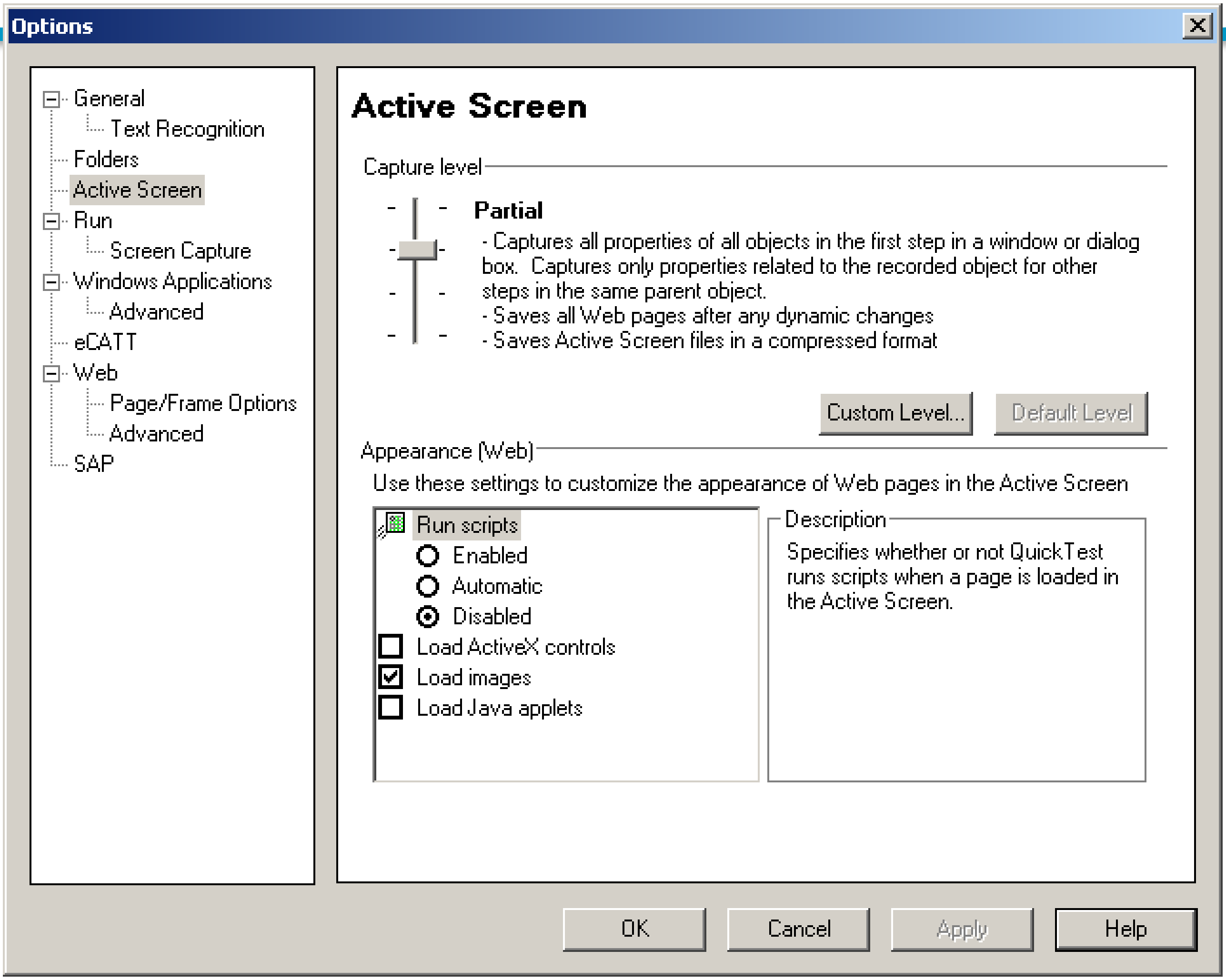
When pointing at a window, activate it after 5 tenths of a second

Restore Layout — Click to restore the panes to their default size and location

Generate Script... — Click to generate an automation script for the global testing options

Tools Option – Active Screen tab

This tab enables to specify which information QuickTest saves and displays in the Active Screen while recording and running tests.
The more information saved in the Active Screen, the easier it is to edit the test after it is recorded.



Active Screen

Capture level

Partial
- Captures all properties of all objects in the first step in a window or dialog box. Captures only properties related to the recorded object for other steps in the same parent object.
- Saves all Web pages after any dynamic changes
- Saves Active Screen files in a compressed format

Appearance (Web)
Use these settings to customize the appearance of Web pages in the Active Screen

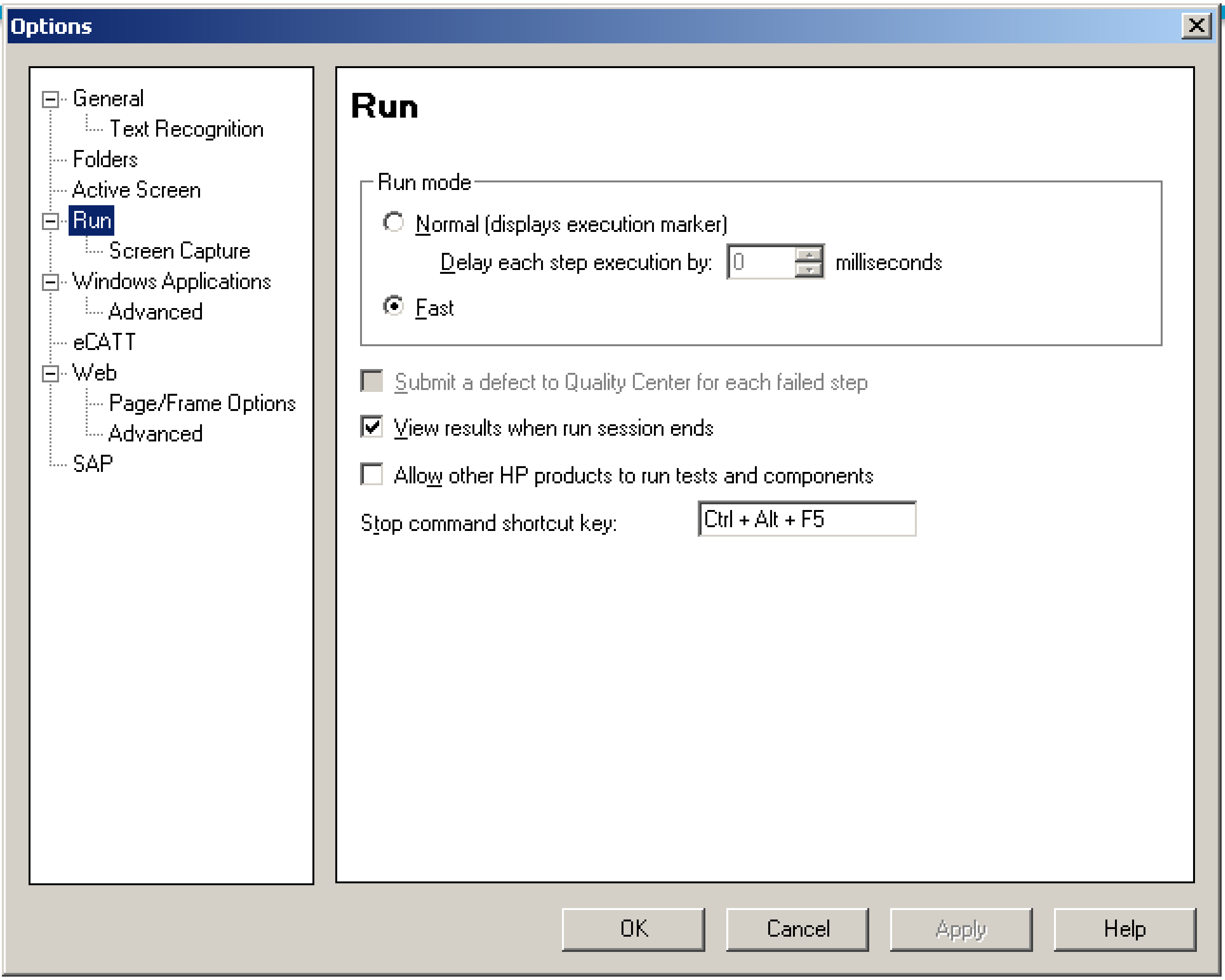
Run scripts
- Enabled
- Automatic
- [✓ Disabled]

- [ ] Load ActiveX controls
- [x] Load images
- [ ] Load Java applets

Description
Specifies whether or not QuickTest runs scripts when a page is loaded in the Active Screen.

Tools Option – Run tab

This tab enables to configure for viewing results after execution is completed, enable test script execution from Quality Center, execution mode and shortcut key for stopping the execution.



Run

Run mode

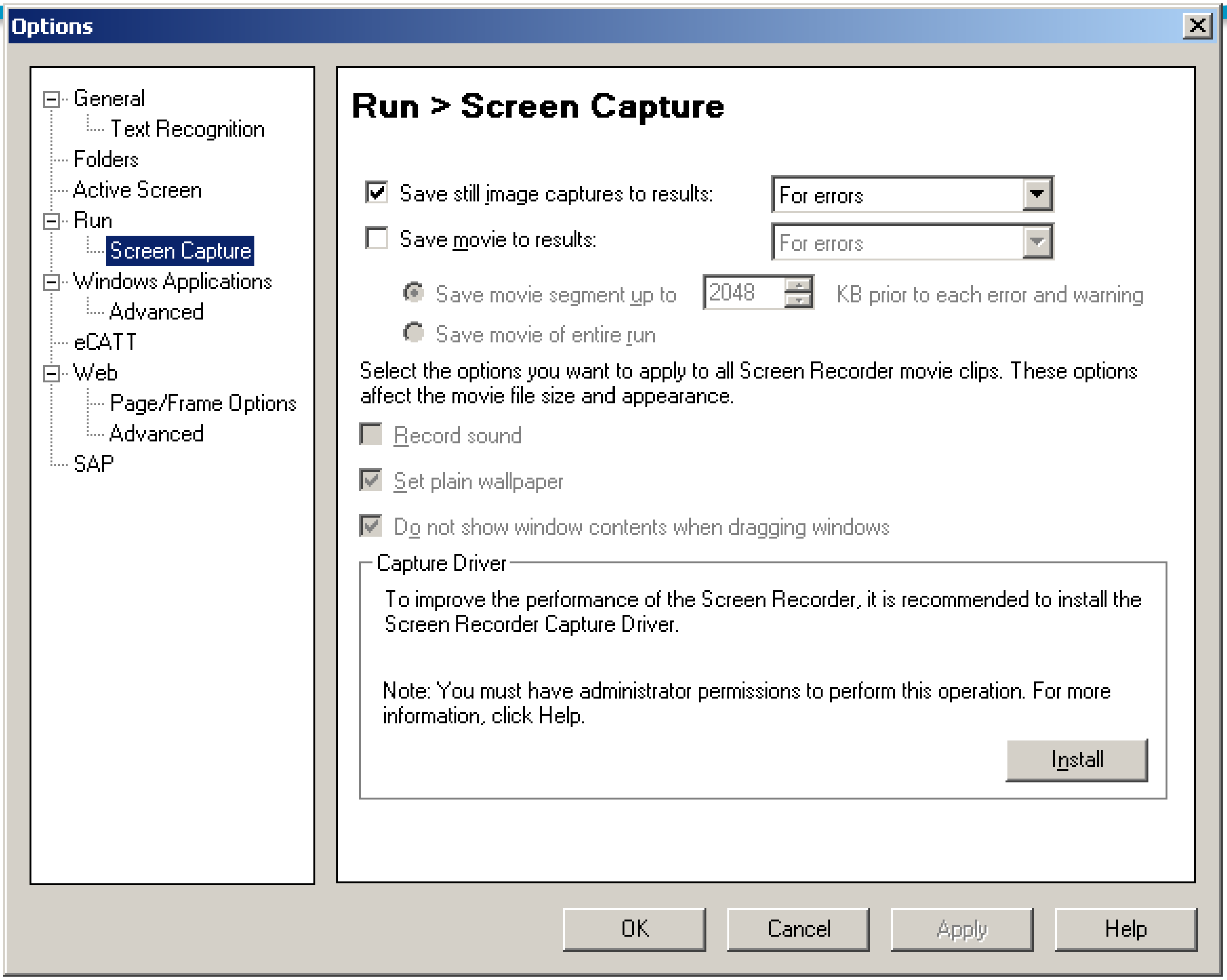
- ( ) Normal (displays execution marker)
  - Delay each step execution by: 0 milliseconds
- (●) Fast

- [ ] Submit a defect to Quality Center for each failed step
- [✓] View results when run session ends
- [ ] Allow other HP products to run tests and components

Stop command shortcut key: Ctrl + Alt + F5

Tools Option – Run->Screen Capture tab

This tab enables to control when and how QuickTest captures screens of the application being tested during the execution.



Run > Screen Capture

- [✓] Save still image captures to results: For errors
- [ ] Save movie to results: For errors
  - (●) Save movie segment up to 2048 KB prior to each error and warning
  - ( ) Save movie of entire run

Select the options you want to apply to all Screen Recorder movie clips. These options affect the movie file size and appearance.

- [ ] Record sound
- [✓] Set plain wallpaper
- [✓] Do not show window contents when dragging windows

Capture Driver

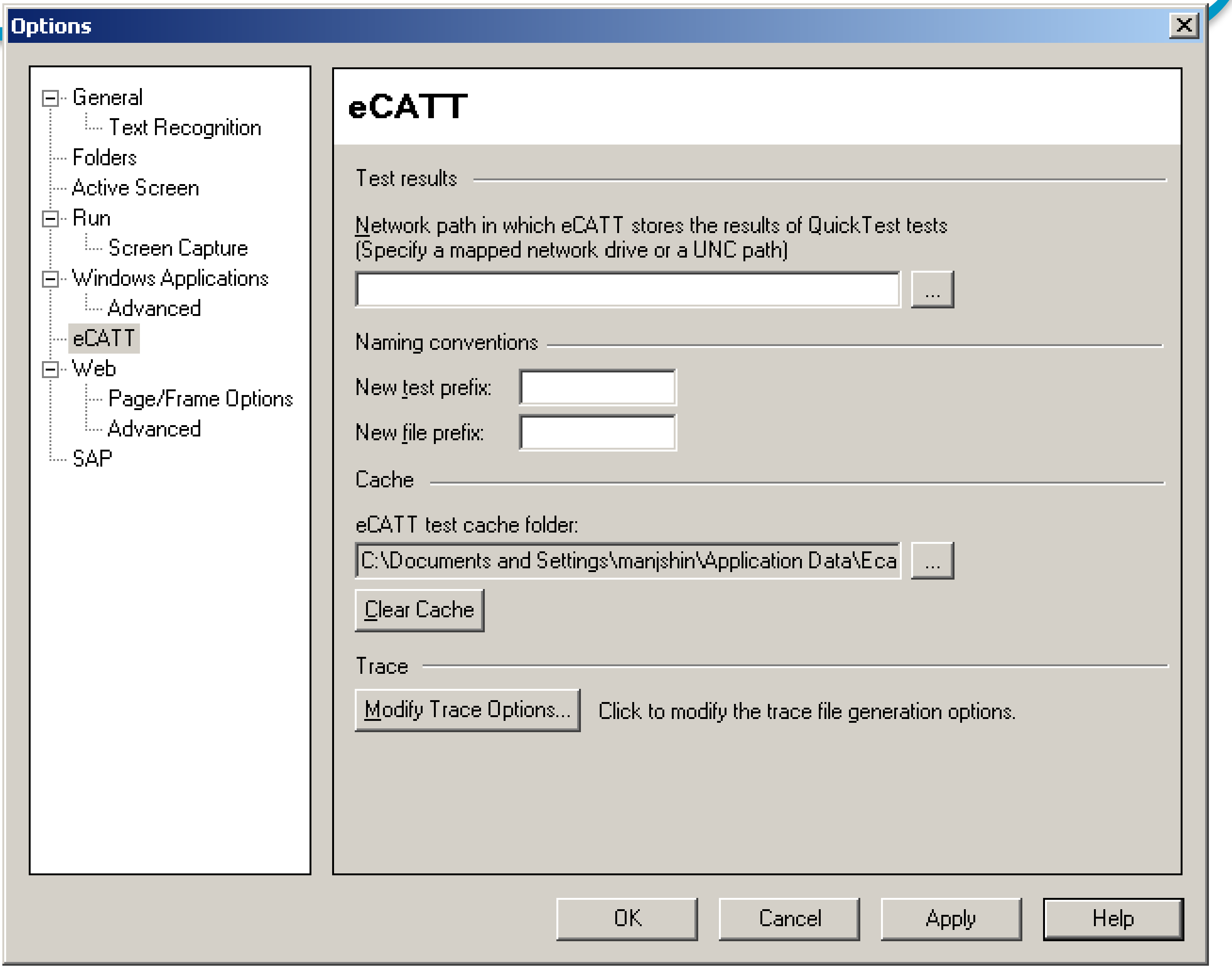
To improve the performance of the Screen Recorder, it is recommended to install the Screen Recorder Capture Driver.

Note: You must have administrator permissions to perform this operation. For more information, click Help.

Tools Option – eCATT tab

The eCATT pane of the Options dialog box enables to configure how QuickTest behaves when connected to eCATT.

The eCATT pane is available only when the QuickTest Professional Add-in for SAP solutions is installed and loaded.



eCATT

Test results

Network path in which eCATT stores the results of QuickTest tests (Specify a mapped network drive or a UNC path)

Naming conventions

New test prefix:

New file prefix:

Cache

eCATT test cache folder: C:\Documents and Settings\manjshin\Application Data\Eca

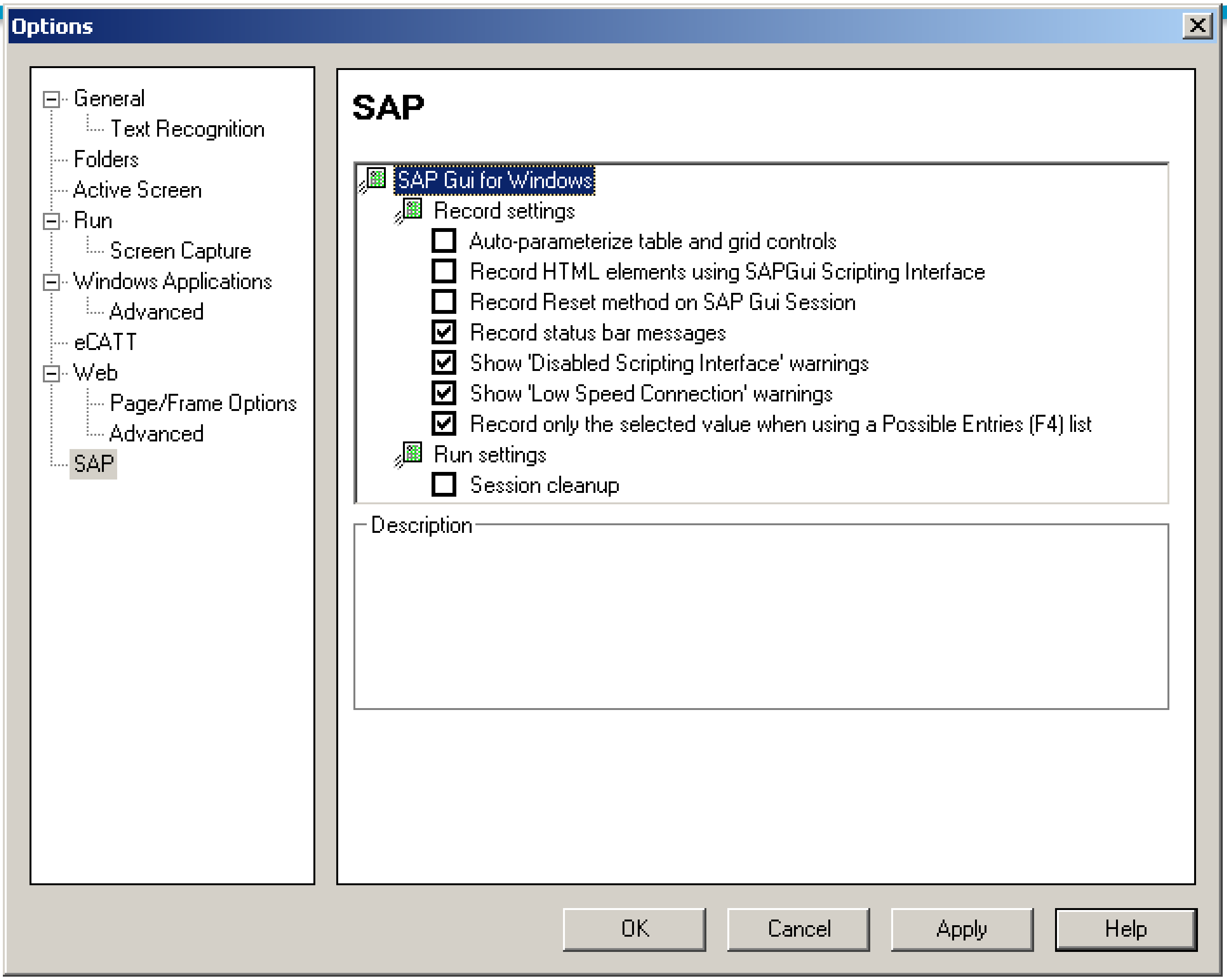
Clear Cache

Trace

Modify Trace Options... Click to modify the trace file generation options.

Tools Option – SAP tab

The SAP pane of the Options dialog box (Tools > Options > SAP node) enables to configure how QuickTest records and runs tests and components on SAP applications



SAP

SAP Gui for Windows

Record settings

- [ ] Auto-parameterize table and grid controls
- [ ] Record HTML elements using SAPGui Scripting Interface
- [ ] Record Reset method on SAP Gui Session
- [✓] Record status bar messages
- [✓] Show 'Disabled Scripting Interface' warnings
- [✓] Show 'Low Speed Connection' warnings
- [✓] Record only the selected value when using a Possible Entries (F4) list

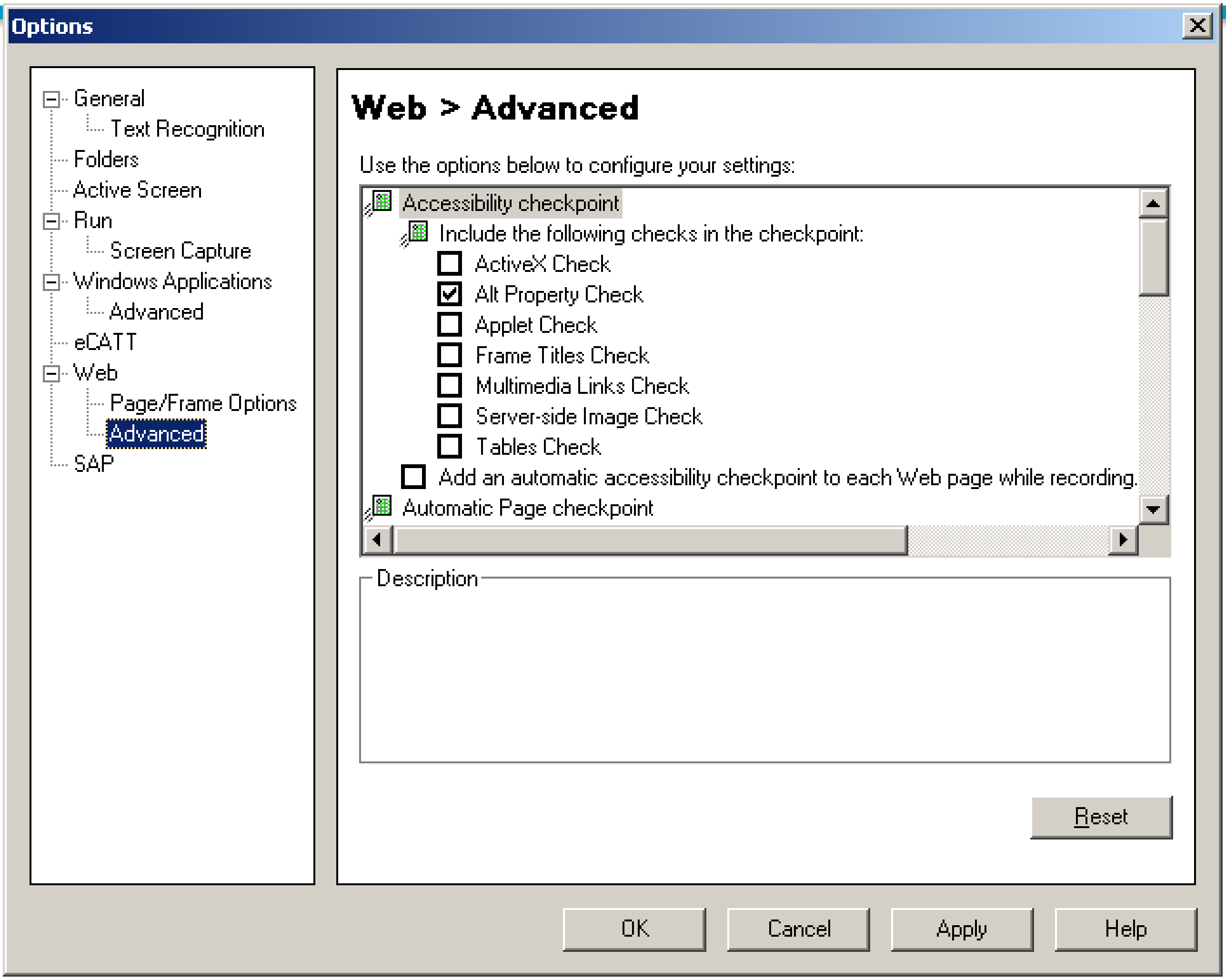
Run settings

- [ ] Session cleanup

Description

Tools Option – Web Advanced tab

When working with tests, we can add accessibility checkpoints to check that Web pages and frames conform to the W3C Web Content Accessibility Guidelines. All accessibility checkpoints in a test use the options that are selected in this dialog box during the run session.



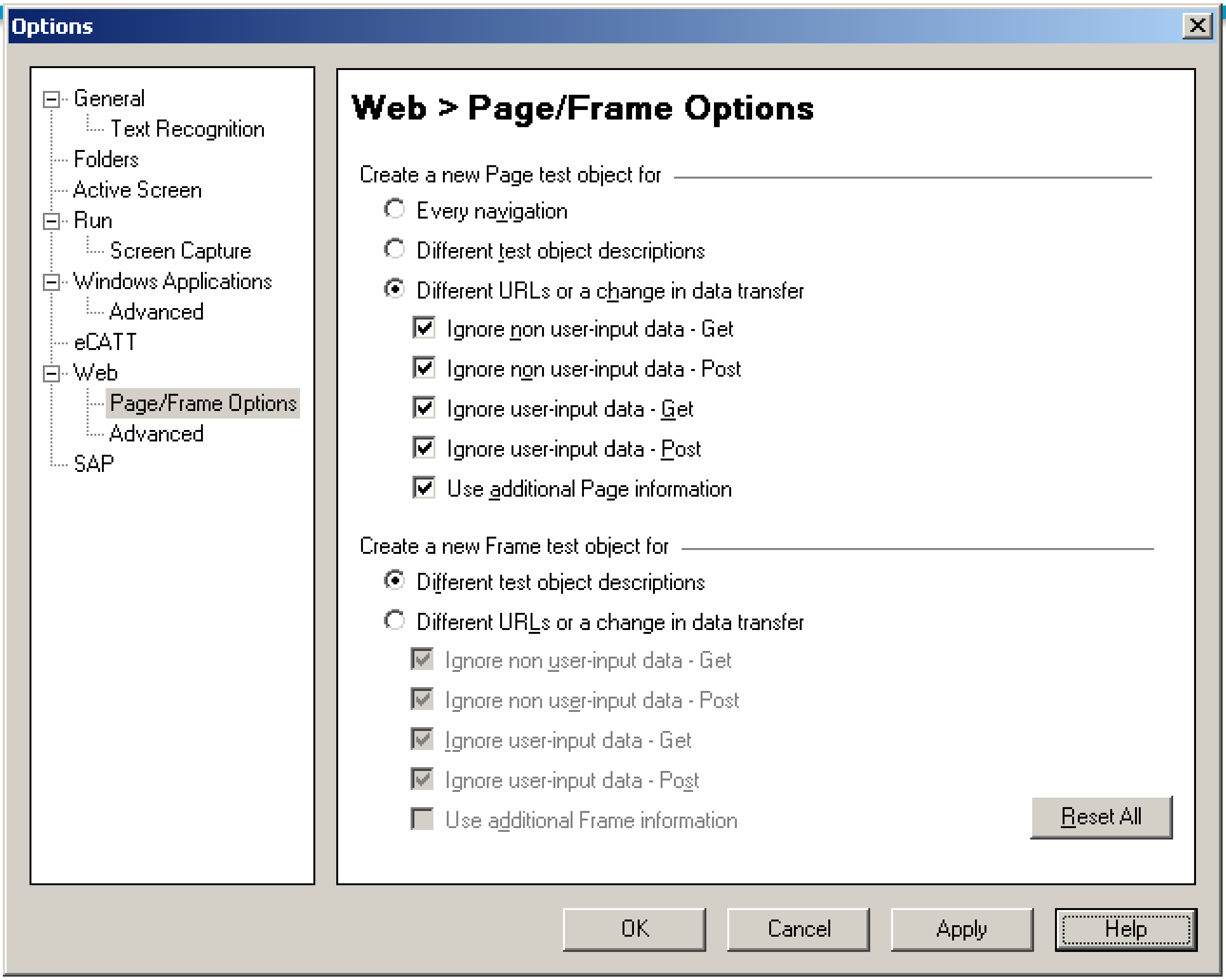
Web > Advanced

Use the options below to configure your settings:

- Accessibility checkpoint
  - Include the following checks in the checkpoint:
    - ☐ ActiveX Check
    - ☑ Alt Property Check
    - ☐ Applet Check
    - ☐ Frame Titles Check
    - ☐ Multimedia Links Check
    - ☐ Server-side Image Check
    - ☐ Tables Check
  - ☐ Add an automatic accessibility checkpoint to each Web page while recording.
- Automatic Page checkpoint

Tools Option – Web tab

The Web > Page/Frame Options pane enables to modify settings for how QuickTest records Page and Frame objects.



Web > Page/Frame Options

Create a new Page test object for

- ○ Every navigation
- ○ Different test object descriptions
- ● Different URLs or a change in data transfer
  - ☑ Ignore non user-input data - Get
  - ☑ Ignore non user-input data - Post
  - ☑ Ignore user-input data - Get
  - ☑ Ignore user-input data - Post
  - ☑ Use additional Page information

Create a new Frame test object for

- ● Different test object descriptions
- ○ Different URLs or a change in data transfer
  - ☑ Ignore non user-input data - Get
  - ☑ Ignore non user-input data - Post
  - ☑ Ignore user-input data - Get
  - ☑ Ignore user-input data - Post
  - ☐ Use additional Frame information

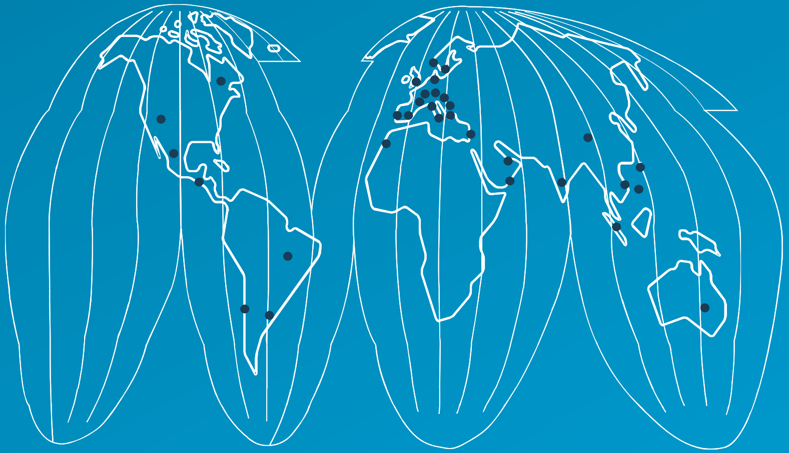
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 120,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2011 global revenues of EUR 9.7 billion.
Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore ®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com