

# OPTIMIZE

**MERCURY QUICKTEST PROFESSIONAL™**

**VERSION 9.2**

**Tutorial**

**MERCURY™**

BUSINESS TECHNOLOGY OPTIMIZATION



# **Mercury QuickTest Professional**

Tutorial

Version 9.2

Document Release Date: February 26, 2007

---

**MERCURY™**

## Mercury QuickTest Professional Tutorial, Version 9.2

This document, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software Documentation is a “commercial item” as defined at 48 C.F.R. 2.101 (October 1995). In accordance with 48 C.F.R. 12.212 (October 1995), 48 C.F.R. 27.401 through 27.404 and 52.227-14 (June 1987, as amended) and 48 C.F.R. 227.7201 through 227.7204 (June 1995), and any similar provisions in the supplements to Title 48 of the C.F.R. (the “Federal Acquisition Regulation”) of other entities of the U.S. Government, as applicable, all U.S. Government users acquire and may use this Documentation only in accordance with the restricted rights set forth in the license agreement applicable to the Computer Software to which this Documentation relates.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Mercury Interactive Corporation  
379 North Whisman Road  
Mountain View, CA 94043  
Tel: (650) 603-5200  
Fax: (650) 603-5300  
<http://www.mercury.com>

© 1992 - 2007 Mercury Interactive Corporation, All rights reserved

If you have any comments or suggestions regarding this document, please send them by e-mail to [documentation@mercury.com](mailto:documentation@mercury.com).

---

# Table of Contents

<b>Welcome to This Tutorial .....</b>	<b>5</b>
How This Tutorial Is Organized .....	6
Who Should Read This Tutorial .....	7
QuickTest Professional Online Documentation .....	8
Additional Online Resources.....	10
Documentation Updates .....	11
Typographical Conventions.....	12
<b>Lesson 1: Introducing QuickTest.....</b>	<b>13</b>
Benefits of Automated Testing .....	13
Testing Process.....	14
QuickTest Window.....	16
Mercury Tours Sample Web Site .....	21
<b>Lesson 2: Recording Tests .....</b>	<b>23</b>
Preparing to Record a Test.....	23
Recording a Test .....	24
Analyzing the Test in the Keyword View.....	30
<b>Lesson 3: Running and Analyzing Tests.....</b>	<b>33</b>
Running a Test.....	33
Analyzing Test Results.....	35
<b>Lesson 4: Creating Checkpoints.....</b>	<b>39</b>
Understanding Checkpoint Types .....	40
Checking Objects .....	42
Checking Pages.....	46
Checking Text .....	48
Checking Tables .....	50
Running and Analyzing a Test with Checkpoints .....	53

<b>Lesson 5: Parameterizing Tests</b> .....	<b>59</b>
Defining a Data Table Parameter .....	60
Adding Parameter Values to a Data Table .....	62
Modifying Steps Affected by Parameterization .....	63
Running and Analyzing a Parameterized Test .....	65
<b>Lesson 6: Creating Output Values</b> .....	<b>69</b>
Creating an Output Value .....	69
Running and Analyzing a Test with Output Values .....	75
<b>Lesson 7: Using Regular Expressions</b> .....	<b>77</b>
Regular Expression Syntax .....	77
Working with Regular Expressions .....	78
Running and Analyzing a Test with Regular Expressions .....	81
<b>Lesson 8: Dividing Tests into Multiple Actions</b> .....	<b>83</b>
Working with Multiple Actions .....	83
Creating New Actions .....	84
Inserting Existing Actions .....	93
Parameterizing an Action .....	99
Running and Analyzing a Multi-Action Test .....	103
<b>Lesson 9: Where Do You Go from Here?</b> .....	<b>105</b>
Testing Your Own Application—Getting Started .....	105
<b>Index</b> .....	<b>109</b>

---

# Welcome to This Tutorial

Welcome to the QuickTest tutorial. The tutorial is a self-paced guide that teaches you the basics of testing your application with QuickTest, the powerful functional testing solution.

This tutorial will familiarize you with the process of creating and running automated tests and analyzing the test results. After completing the tutorial, you can apply the skills you have learned to your own application or Web site.

---

**Note:** In addition to tests, QuickTest enables you to create business components for use in business process tests, if you have Quality Center with Business Process Testing support installed. The procedures described in this tutorial are designed for creating tests, but you can also apply the majority of these procedures to creating business components. For more information on business components and Business Process Testing, refer to the *Mercury QuickTest Professional for Business Process Testing User's Guide* and the *Mercury Business Process Testing User's Guide*.

---

<b>This chapter describes:</b>	<b>On page:</b>
How This Tutorial Is Organized	6
Who Should Read This Tutorial	7
QuickTest Professional Online Documentation	8
Additional Online Resources	10
Documentation Updates	11
Typographical Conventions	12

## **How This Tutorial Is Organized**

The tutorial is divided into short lessons. In each lesson you will create and run tests on the sample Mercury Tours Web site.

This tutorial contains the following lessons:

### **Lesson 1 Introducing QuickTest**

Compares automated and manual testing methods. It introduces the QuickTest testing process and familiarizes you with the QuickTest user interface and the sample Mercury Tours Web site.

### **Lesson 2 Recording Tests**

Teaches you how to record a test and provides an overview of the resulting Keyword View display.

### **Lesson 3 Running and Analyzing Tests**

Describes how to run a test and view the test results.

### **Lesson 4 Creating Checkpoints**

Provides an overview of different checkpoint types and explains how to add checkpoints to your test to verify that information in your Web site is displayed as expected.



**Lesson 5    Parameterizing Tests**

Describes how to parameterize a test using Data Table parameters, and then run the test on multiple sets of data and analyze the results.

**Lesson 6    Creating Output Values**

Describes how to use output parameters to retrieve data from the sample Web site, and then run the test and analyze its results.

**Lesson 7    Using Regular Expressions**

Teaches you how to create and run a test using regular expressions.

**Lesson 8    Dividing Tests into Multiple Actions**

Explains how to divide your test into actions so that you can design more efficient and modular tests.

**Lesson 9    Where Do You Go from Here?**

Shows you how to get started testing your own application.

## **Who Should Read This Tutorial**

This tutorial is intended for users who are new to QuickTest. No prior knowledge of QuickTest is required. A general understanding of testing concepts and functional testing processes may be helpful, but is not mandatory.

## QuickTest Professional Online Documentation

QuickTest Professional includes the following online documentation:

**Readme** provides the latest news and information about QuickTest. Choose **Start > Programs > QuickTest Professional > Readme**.

**QuickTest Professional Installation Guide** explains how to install and set up QuickTest. Choose **Help > Printer-Friendly Documentation > Mercury QuickTest Professional Installation Guide**.

**QuickTest Professional Tutorial** teaches you basic QuickTest skills and shows you how to design tests for your applications. Choose **Help > QuickTest Professional Tutorial**.

**Product Feature Movies** provide an overview and step-by-step instructions describing how to use selected QuickTest features. Choose **Help > Product Feature Movies**.

**Printer-Friendly Documentation** displays the complete documentation set in Adobe portable document format (PDF). Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site (<http://www.adobe.com>). Choose **Help > Printer-Friendly Documentation**.

**QuickTest Professional Help** includes:

- **What's New in QuickTest Professional** describes the newest features, enhancements, and supported environments in the latest version of QuickTest.
- **QuickTest User's Guide** describes how to use QuickTest to test your application.
- **QuickTest for Business Process Testing User's Guide** provides step-by-step instructions for using QuickTest to create and manage assets for use with Business Process Testing.
- **QuickTest Object Model** describes QuickTest test objects, lists the methods and properties associated with each object, and provides syntax information and examples for each method and property.

- ▶ **QuickTest Advanced References** contains documentation for the following QuickTest COM and XML references:
  - ▶ **QuickTest Automation** provides syntax, descriptive information, and examples for the automation objects, methods, and properties. It also contains a detailed overview to help you get started writing QuickTest automation scripts. The automation object model assists you in automating test management, by providing objects, methods and properties that enable you to control virtually every QuickTest feature and capability.
  - ▶ **QuickTest Test Results Schema** documents the XML schema that enables you to customize your test results.
  - ▶ **QuickTest Test Object Schema** documents the XML schema that enables you to extend test object support in different environments.
  - ▶ **QuickTest Object Repository Automation** documents the Object Repository automation object model that enables you to manipulate QuickTest object repositories and their contents from outside of QuickTest.
- ▶ **VBScript Reference** contains Microsoft VBScript documentation, including VBScript, Script Runtime, and Windows Script Host.

To access the QuickTest Professional Help, choose **Help > QuickTest Professional Help**. You can also access the QuickTest Professional Help by clicking in selected QuickTest windows and dialog boxes and pressing F1. Additionally, you can view a description, syntax, and examples for a QuickTest test object, method, or property by placing the cursor on it and pressing F1.

---

**Note:** Your QuickTest Help may contain additional items relevant to any QuickTest add-ins you have installed. For more information, refer to the relevant add-in documentation.

---

## Additional Online Resources

**Mercury Tours** sample Web site is the basis for many examples in this guide. The URL for this Web site is <http://newtours.mercury.com>. Choose **Start > Programs > QuickTest Professional > Sample Applications > Mercury Tours Web Site**.

**Knowledge Base** uses your default Web browser to open the Mercury Customer Support Web site directly to the Knowledge Base landing page. Choose **Help > Knowledge Base**. The URL for this Web site is <http://support.mercury.com/cgi-bin/portal/CSO/kbBrowse.jsp>.

**Customer Support Web Site** uses your default Web browser to open the Mercury Customer Support Web site. This site enables you to browse the Mercury Support Knowledge Base and add your own articles. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > Customer Support Web Site**. The URL for this Web site is <http://support.mercury.com>.

**Send Feedback** enables you to send online feedback about **QuickTest Professional** to the product team. Choose **Help > Send Feedback**.

**Mercury Home Page** uses your default Web browser to access the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > Mercury Home Page**.

**Mercury Best Practices** contain guidelines for planning, creating, deploying, and managing a world-class IT environment. Mercury provides three types of best practices: Process Best Practices, Product Best Practices, and People Best Practices. Licensed customers of Mercury software can read and use the Mercury Best Practices available from the Customer Support site, <http://support.mercury.com>.

## Documentation Updates

Mercury is continually updating its product documentation with new information. You can download the latest version of this document from the Customer Support Web site (<http://support.mercury.com>).

### To download updated documentation:

- 1** If you are not already logged into the Customer Support Web site, enter your login information and click **Login**.
- 2** Click the **Documentation** link.
- 3** Under **Please Select Product**, select **QuickTest Professional**.

Note that if the required product does not appear in the list, you must add it to your customer profile. Click **My Account** to update your profile.

- 4** Click **Retrieve**. The Documentation page opens and lists the documentation available for the current release and for previous releases. If a document was updated recently, **Updated** appears next to the document name.
- 5** Click a document link to download the documentation.

## Typographical Conventions

This guide uses the following typographical conventions:

<b>UI Elements and Function Names</b>	This style indicates the names of interface elements on which you perform actions, file names or paths, and other items that require emphasis. For example, “Click the <b>Save</b> button.” It also indicates method or function names. For example, “The <b>wait_window</b> statement has the following parameters:”
<i>Arguments</i>	This style indicates method, property, or function arguments and book titles. For example, “Refer to the <i>Mercury User’s Guide</i> .”
<b>&lt;Replace Value&gt;</b>	Angle brackets enclose a part of a file path or URL address that should be replaced with an actual value. For example, <b>&lt;MyProduct installation folder&gt;\bin</b> .
Example	This style is used for examples and text that is to be typed literally. For example, “Type Hello in the edit box.”
CTRL+C	This style indicates keyboard keys. For example, “Press ENTER.”
[ ]	Square brackets enclose optional arguments.
{ }	Curly brackets indicate that one of the enclosed values must be assigned to the current argument.
...	In a line of syntax, an ellipsis indicates that more items of the same format may be included. In a programming example, an ellipsis is used to indicate lines of a program that were intentionally omitted.
	A vertical bar indicates that one of the options separated by the bar should be selected.

# 1

---

## Introducing QuickTest

This lesson provides you with an overview of automated testing and QuickTest, and gets you set up to perform the tutorial.

In this lesson, you will learn about:	On page:
Benefits of Automated Testing	13
Testing Process	14
QuickTest Window	16
Mercury Tours Sample Web Site	21

### Benefits of Automated Testing

If you have ever tested applications or Web sites manually, you are aware of the drawbacks. Manual testing is time-consuming and tedious, requiring a heavy investment in human resources. Worst of all, time constraints often make it impossible to manually test every feature thoroughly before the application is released. This leaves you wondering whether serious bugs have gone undetected.

Automated testing with QuickTest addresses these problems by dramatically speeding up the testing process. You can create tests that check all aspects of your application or Web site, and then run these tests every time your site or application changes.

As QuickTest runs tests, it simulates a human user by moving the cursor in a Web page or application window, clicking GUI (graphical user interface) objects, and entering keyboard input; however, QuickTest does this faster than any human user.

<b>Benefits of Automated Testing</b>	
Fast	QuickTest runs tests significantly faster than human users.
Reliable	Tests perform precisely the same operations each time they are run, thereby eliminating human error.
Repeatable	You can test how the Web site or application reacts after repeated execution of the same operations.
Programmable	You can program sophisticated tests that bring out hidden information.
Comprehensive	You can build a suite of tests that covers every feature in your Web site or application.
Reusable	You can reuse tests on different versions of a Web site or application, even if the user interface changes.

## Testing Process

The QuickTest testing process consists of seven main phases:

### 1 Preparing to record

Before you record a test, confirm that your application and QuickTest are set to match the needs of your test.

Make sure your application displays elements on which you want to record, such as a toolbar or a special window pane, for example, and that your application options are set as you expect for the purposes of your test.



## **2 Recording a session on your application**

As you navigate through your application or Web site, QuickTest graphically displays each step you perform as a row in the Keyword View. A step is any user action that causes or makes a change in your application, such as clicking a link or image, or entering data in a form.

## **3 Enhancing your test**

- ▶ Inserting checkpoints into your test lets you search for a specific value of a page, object, or text string, which helps you determine whether your application or site is functioning correctly.
- ▶ Broadening the scope of your test, by replacing fixed values with parameters, lets you check how your application performs the same operations with multiple sets of data.
- ▶ Adding logic and conditional or loop statements enables you to add sophisticated checks to your test.

## **4 Debugging your test**

You debug a test to ensure that it operates smoothly and without interruption.

## **5 Running your test**

You run a test to check the behavior of your application or Web site. While running, QuickTest opens the application, or connects to the Web site, and performs each step in your test.

## **6 Analyzing the test results**

You examine the test results to pinpoint defects in your application.

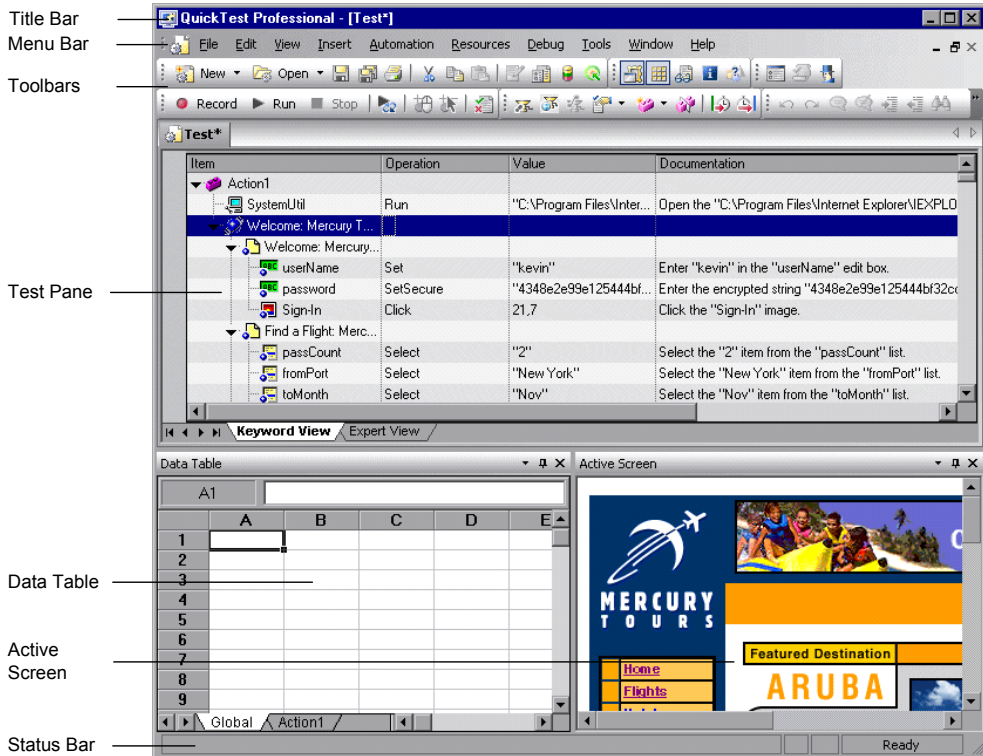
## **7 Reporting defects**

If you have Quality Center installed, you can report the defects you discover to a database. Quality Center is the Mercury test management solution.

## QuickTest Window

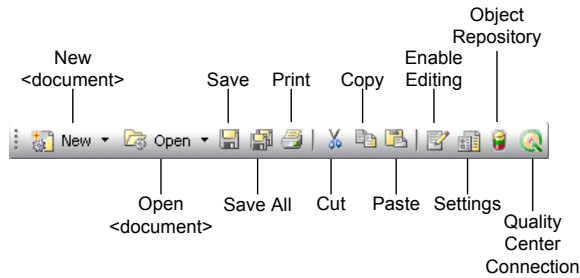
Before you begin creating tests, you should familiarize yourself with the main QuickTest window.

The image below shows a QuickTest window as it would appear after you record a test, with all toolbars and the Data Table and Active Screen panes displayed:

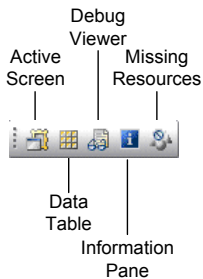


The QuickTest window contains the following key elements:

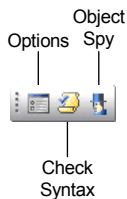
- **Title bar.** Displays the name of the currently open test.
- **Menu bar.** Displays menus of QuickTest commands.
- **Standard toolbar.** Contains buttons to assist you in managing your test.



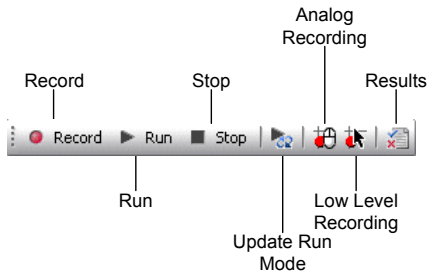
- **View toolbar.** Contains buttons to view the various panes that can assist you in the testing process.



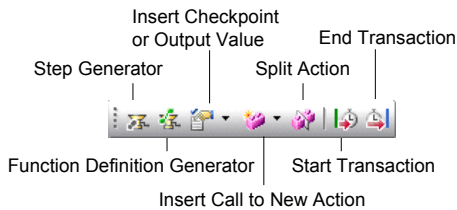
- **Tools toolbar.** Contains buttons to assist you in setting options, checking syntax, and working with the Object Spy.



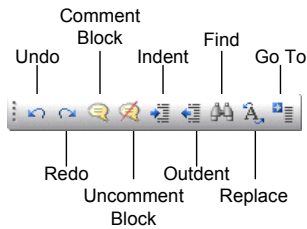
- ▶ **Automation toolbar.** Contains buttons to assist you in the testing process.



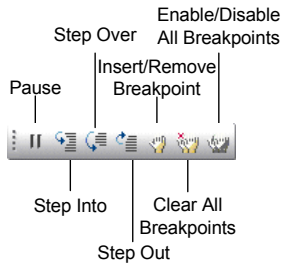
- ▶ **Insert toolbar.** Contains buttons to assist you in inserting steps, actions, checkpoints, and output values into your test.



- ▶ **Edit toolbar.** Contains buttons to assist you in editing your test steps.



- **Debug toolbar.** Contains buttons to assist you in debugging tests.





**Note:** The Debug toolbar is not displayed when you open QuickTest for the first time. You can display the Debug toolbar by choosing **View > Toolbars > Debug**, or by clicking the **Debug Viewer** button. This tutorial does not describe how to debug a test. For additional information, refer to the *Mercury QuickTest Professional User's Guide*.

- **Action toolbar.** Contains buttons and a list of actions, enabling you to view the details of an individual action or the entire test flow.



**Note:** The Action toolbar is displayed only in the Keyword View. It is not displayed when you open QuickTest for the first time. You can display the Action toolbar by choosing **View > Toolbars > Action**. If you insert a reusable or external action in a test, the Action toolbar is displayed automatically in the Keyword View. For additional information, refer to the *QuickTest Professional User's Guide*.

- ▶ **Test pane.** Contains the Keyword View and Expert View tabs.
- ▶ **Active Screen.** Provides a snapshot of your application as it appeared when you performed a certain step during the recording session.
- ▶ **Data Table.** Assists you in parameterizing your test.
-  ▶ **Debug Viewer pane.** Assists you in debugging your test. The Debug Viewer pane contains the **Watch**, **Variables**, and **Command** tabs. (The Debug Viewer pane is not displayed when you open QuickTest for the first time. You can display the Debug Viewer by choosing **View > Debug Viewer**, or by clicking the **Debug Viewer** button.)
- ▶ **Information pane.** Displays a list of syntax errors found in your test.
-  ▶ **Missing Resources pane.** Provides a list of the resources that are specified in your test but cannot be found, such as unmapped shared object repositories and parameters that are connected to shared object repositories. (The Missing Resources pane is not displayed when you open QuickTest for the first time. You can display the Missing Resources pane by choosing **View > Missing Resources**, or by clicking the **Missing Resources** button.)
- ▶ **Status bar.** Displays the status of the QuickTest application and other relevant information.

Some of the panes and toolbar options listed above are described in detail in subsequent lessons. For information on other panes and toolbar options, refer to the *QuickTest Professional User's Guide*.

You can also execute many menu commands using shortcut keys. For additional information, refer to the "QuickTest at a Glance" chapter in the *QuickTest Professional User's Guide*.

Now that you are familiar with the main QuickTest window, take a few minutes to explore its components.

## Mercury Tours Sample Web Site

Mercury Tours is the sample Web application used throughout this tutorial. It simulates a Web-based flight information and reservation service. You should familiarize yourself with this application before starting the tutorial. You can access the Mercury Tours Web site using any Web browser, but you can perform the tutorial using only Microsoft Internet Explorer.

### Optimizing Browser Settings for Your Test

In Microsoft Internet Explorer, you should clear the option to use AutoComplete for user names and passwords for the purposes of this tutorial. This ensures that all of your operations are accurately recorded while creating your tests.

**To clear the AutoComplete option:**

- 1** In the Microsoft Internet Explorer menu bar, choose **Tools > Internet Options > Content** tab.
- 2** Click **AutoComplete** in the **Personal information** area. The AutoComplete Settings dialog box opens.
- 3** In the **Use AutoComplete for** area, clear the **User names and passwords on forms** option.
- 4** Click **OK** to save your changes and close the AutoComplete Settings dialog box, then click **OK** again to close the Internet Options dialog box.

### Using the Mercury Tours Web Site for the First Time

Before you begin recording your tests on the Mercury Tours Web site, you must register as a user.

**To run Mercury Tours:**

- 1** **Launch the Mercury Tours application.**

In Microsoft Internet Explorer, type the following URL:  
<http://newtours.mercury.com>

The Mercury Tours home page opens.

## 2 Register with Mercury Tours.

To log on and use the Mercury Tours application, you must be a registered user.

From the home page, click the **REGISTER** navigation link. The Register page opens.

In the User Information section at the bottom of the page, enter any user name and password, and confirm your password. (Other information is not required.)

Click **SUBMIT**. Mercury Tours confirms your registration. From this confirmation page, click the **sign-in** navigation link to enter the application. The SIGN-ON page opens. Enter your registered user name and password, and click **SUBMIT**.

The Flight Finder page opens.

## 3 Explore the Mercury Tours site.

Beginning with the Flight Finder page, follow the on-screen instructions to get flight information and reserve a flight.

On the Book a Flight page, you do not have to fill out actual personal or credit card information; just enter the required information (indicated by red) in the Passengers and Credit Card sections. (You may enter fictitious information.)

## 4 End your Mercury Tours session.

When you have finished exploring the Mercury Tours application, click the **LOG OUT** button on the Flight Confirmation page or the **SIGN-OFF** link at the top of any application page.

To sign on again, simply enter your registered user name and password in the Sign on page or in the Mercury Tours home page.

## 5 Close the Web browser.

You are now ready to begin using this tutorial to create tests on the Mercury Tours application using QuickTest!



# 2

---

## Recording Tests

As you navigate through your Web site or application, QuickTest records your steps. These operations are the basis of your test. When you stop recording, you can see the steps of your test in the Keyword View.

In this lesson, you will learn about:	On page:
Preparing to Record a Test	23
Recording a Test	24
Analyzing the Test in the Keyword View	30

### Preparing to Record a Test

Before you begin recording a test, ensure that both your application or Web site and QuickTest are set to match the needs of your test. For the purposes of this tutorial, ensure that:

- ▶ You have registered as a user in the Mercury Tours Web site. For more information, see “Using the Mercury Tours Web Site for the First Time” on page 21.
- ▶ In Microsoft Internet Explorer, you have cleared the **AutoComplete** option for user names and passwords. For instructions, see “Optimizing Browser Settings for Your Test” on page 21.
- ▶ All browsers are closed before you begin recording.
- ▶ The proper Web Page/Frame options are set. Open QuickTest, choose **Tools > Options** and click the **Web** tab. Click the **Page/Frame Options** button. In the **Create a new Page test object for** area, select **Different URLs or a change in data transfer** and select the first four check boxes.

## Recording a Test

In this section, you will record the process of making a reservation for a flight from New York to San Francisco on the Mercury Tours Web site.



### 1 Start QuickTest and open a new test.

- If QuickTest is not currently open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

In the Add-in Manager, confirm that the **Web** Add-in is selected, and clear all other add-ins. Click **OK** to close the Add-in Manager and open QuickTest.

---

**Note:** While QuickTest loads your selected add-ins, the QuickTest splash screen is displayed. This may take a few seconds.

---

If the Welcome window opens, click **Blank Test**.



Otherwise, choose **File > New > Test**, or click the **New** button down arrow and choose **Test**.

---

**Note:** If a test is already open, clicking the **New** toolbar button opens a new test.

---

A blank test opens.

- If QuickTest is already open, check which add-ins are loaded by selecting **Help > About QuickTest Professional**. If the **Web** Add-in is not loaded, you must exit and restart QuickTest. When the Add-in Manager opens, select the **Web** Add-in, and clear all other add-ins.

---

**Note:** If the Add-in Manager does not open when starting QuickTest, choose **Tools > Options**. In the General tab, select **Display Add-in Manager on startup**. When you exit and restart QuickTest, the Add-in Manager opens.

---

## 2 Start recording on the Mercury Tours Web site.

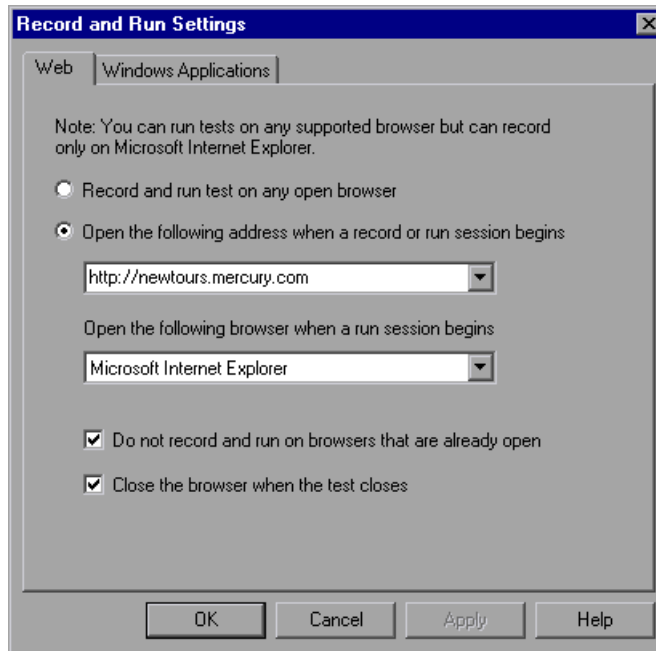


Choose **Automation > Record** or click the **Record** button. The Record and Run Settings dialog box opens.

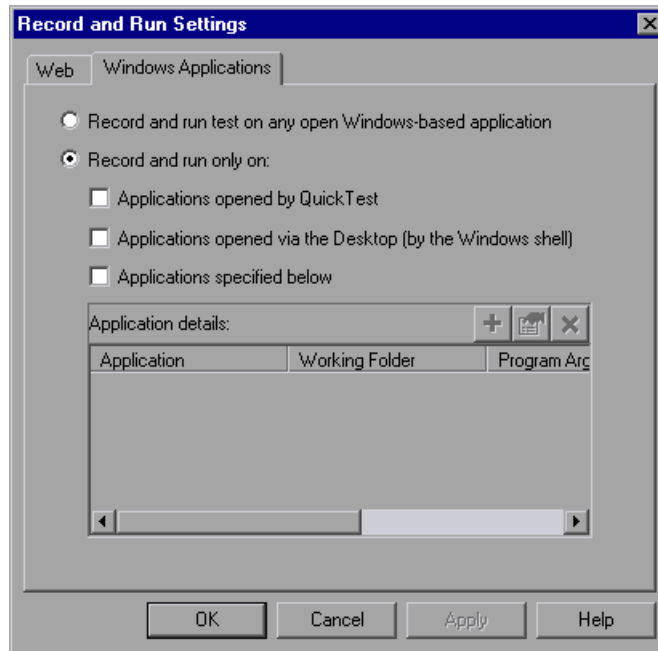
- ▶ In the Web tab, select **Open the following address when a record or run session begins**.

Confirm that the URL in the first box is <http://newtours.mercury.com> and in the second box, choose a browser on which to run your test session. Microsoft Internet Explorer is the browser used in this tutorial.

Confirm that **Do not record and run on browsers that are already open** and **Close the browser when the test closes** are selected.



- In the Windows Applications tab, confirm that **Record and run only on** is selected, and make sure that all three check boxes are cleared.



These settings prevent you from inadvertently recording operations performed on various Windows applications (such as e-mail) during a recording session.

Click **OK**.

QuickTest begins recording, and your browser opens to the Mercury Tours Web site.

### 3 Log in to the Mercury Tours Web site.

In the **User Name** and **Password** boxes, type the name and password you registered with Mercury Tours.

Click **Sign-In**.

The Flight Finder page opens.

#### 4 Enter flight details.

Change the following selections:

- Departing From: **New York**
- On: **December 29**
- Arriving In: **San Francisco**
- Returning: **December 31**
- Service Class: **Business class**

Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

---

#### Notes:

When selecting the date, you must click the drop-down list, scroll to any new entry, and select it. This is because QuickTest only records a step if the value in the list changes. The tutorial will not work correctly if you accept the currently displayed month or day.

When entering dates while recording this test, do not click the **View Calendar** button, which opens a Java-based calendar. Your test will not record the date selected using this calendar because the QuickTest Java Add-in is not loaded for this tutorial.

To check which add-ins are loaded, choose **Help > About QuickTest Professional**. To change the available add-ins for your tests, you must close and reopen QuickTest Professional.

---

#### 5 Select a flight.

Click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

## 6 Enter required passenger and purchase information.

Enter the required information (fields with red text labels) in the Passengers and Credit Card sections. (You may enter fictitious information.)

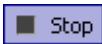
In the Billing Address section, select **Ticketless Travel**.

At the bottom of the page, click **SECURE PURCHASE**. The Flight Confirmation page opens.

## 7 Review and complete your booking.

Click **BACK TO HOME**. The Mercury Tours home page opens.

## 8 Stop recording.



In QuickTest, click **Stop** on the Testing toolbar to stop the recording process.

You have now reserved an imaginary business class ticket from New York to San Francisco. QuickTest recorded your Web browser operations from the time you clicked the **Record** button until you clicked the **Stop** button.

## 9 Save your test.



Choose **File > Save** or click the **Save** button. The Save dialog box opens to the Tests folder.

Create a folder named Tutorial, select it, and click **Open**.

Type Recording in the **File name** field.

Confirm that **Save Active Screen files** is selected.

Click **Save**. The test name (Recording) and path are displayed in the title bar of the main QuickTest window.

---

**Note:** If a test has changed and is not yet saved, the test name in the title bar contains an asterisk.

---

## Analyzing the Test in the Keyword View

As you recorded your test, QuickTest generated steps in the Keyword View representing each operation you performed in the Web browser.

The columns in the Keyword View show different information for each step, as follows:

- ▶ **Item.** The item for the step (test object, utility object, function call, or statement) in a hierarchical icon-based tree.
- ▶ **Operation.** The operation to be performed on the item, for example, **Click** or **Select**.
- ▶ **Value.** The argument values for the selected operation, for example, the mouse button to use when clicking the image.
- ▶ **Documentation.** Auto-documentation of what the step does, in an easy-to-understand sentence, for example, Click the "findFlights" image.
- ▶ **Assignment.** The assignment of a value to or from a variable so you can use the value later in the test. This column is not visible by default.
- ▶ **Comment.** Any textual information you want to add regarding the step, for example, Return to page used in first step of the test. This column is not visible by default.

---

**Note:** You can choose to hide or display individual columns as required, by right-clicking the column heading in the Keyword View and selecting a column name from the list.

---

In the **Item** column of the Keyword View, you can click the branch arrows to expand or collapse the steps under each Web page. You can expand the entire test by choosing **View > Expand All**.









When you expand your test, it should look similar to this:

Item	Operation	Value	Documentation
▼ Action1			
▼ Welcome: Mercury Tours			
▼ Welcome: Mercury Tours			
userName	Set	"mercury"	Enter "mercury" in the "userName" edit box.
password	SetSecure	"4082820183...	Enter the encrypted string "4082820183afe512e8bc91c1f7222db...
Sign-In	Click	2,2	Click the "Sign-In" image.
▼ Find a Flight: Mercury			
fromPort	Select	"New York"	Select item "New York" in the "fromPort" list.
fromMonth	Select	"Dec"	Select item "Dec" in the "fromMonth" list.
fromDay	Select	"29"	Select item "29" in the "fromDay" list.
toPort	Select	"San Francisco"	Select item "San Francisco" in the "toPort" list.
toMonth	Select	"Dec"	Select item "Dec" in the "toMonth" list.
toDay	Select	"31"	Select item "31" 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.
▼ Select a Flight: Mercury			
reserveFlights	Click	2,2	Click the "reserveFlights" image.
▼ Book a Flight: Mercury			
passFirst0	Set	"Nicole"	Enter "Nicole" in the "passFirst0" edit box.
passLast0	Set	"Jones"	Enter "Jones" in the "passLast0" edit box.
creditnumber	Set	"12345"	Enter "12345" in the "creditnumber" edit box.
ticketLess	Set	"ON"	Set the state of the "ticketLess" check box to "ON".
buyFlights	Click	2,2	Click the "buyFlights" image.
▼ Flight Confirmation: Mercury			
home	Click		Click the "home" image.
Welcome: Mercury Tours	Sync		Wait for the Web page to synchronize before continuing the run.

For every step performed on an object, QuickTest displays a row in the Keyword View with an icon and details of the step. For example, the step below instructs QuickTest to click the **reserveFlights** image at a specific location in the **Select a Flight: Mercury** page of the Mercury Tours Web site.

Item	Operation	Value	Documentation
▼ Select a Flight: Mercury			
reserveFlights	Click	2,2	Click the "reserveFlights" image.

The table below provides an explanation of the first few steps shown in the test in the Keyword View above.

Step	Description
 Action1	<b>Action1</b> is the action name. Actions are test units that divide your test into logical sections.
 Welcome: Mercury Tours	The browser invokes the <b>Welcome: Mercury Tours</b> Web site.
 Welcome: Mercury Tours	<b>Welcome: Mercury Tours</b> is the name of the Web page.
 userName    Set    "mercury"	<b>userName</b> is the name of the edit box. <b>Set</b> is the method performed on the edit box. <b>mercury</b> is the value of the edit box.
 password    SetSecure    "4082820183..."	<b>password</b> is the name of the edit box. <b>SetSecure</b> is an encryption method performed on the edit box. <b>4082820183afe512e8bc91c1f7222dbd</b> is the encrypted value of the password.
 Sign-In    Click    2,2	<b>Sign-In</b> is the name of the image link. <b>Click</b> is the method performed on the image. <b>2, 2</b> are the x- and y-coordinates where the image was clicked.

Take a few minutes to examine the Keyword View in detail, tracing the steps you performed while recording your test.

For more information on the Keyword View, refer to the *Mercury QuickTest Professional User's Guide*.

# 3

---

## Running and Analyzing Tests

When you run your test, QuickTest opens the appropriate application or Web site and performs each step as it was originally recorded in the test. When QuickTest finishes running the test, it displays the results of the run.

In this lesson, you will learn about:	On page:
Running a Test	33
Analyzing Test Results	35

### Running a Test

In this lesson, you will run the test you recorded in the previous lesson.



#### 1 Start QuickTest and open the Recording test.

If QuickTest is not already open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

► If the Welcome window opens, click **Open Existing**.



► If QuickTest opens without displaying the Welcome window, choose **File > Open > Test** or click the **Open** button down arrow and choose **Test**.

In the Open Test dialog box, locate and select the **Recording** test, then click **Open**.

## 2 Confirm that all images are saved to the test results.

QuickTest allows you to determine when to save images to the test results. In this lesson, all images should be saved to the test results.



Choose **Tools > Options** or click the **Options** button, then click the **Run** tab. In the **Save still image captures to results** option, select **Always**. Click **OK** to close the Options dialog box.

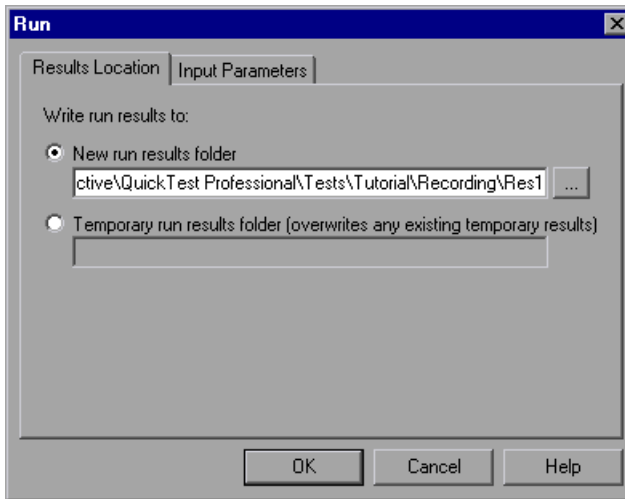
## 3 Start running your test.



Click **Run** or choose **Automation > Run**. The Run dialog box opens.

Select **New run results folder**. Accept the default results folder name.

Click **OK** to close the Run dialog box.



Watch carefully as QuickTest opens your browser and starts running the test. In the browser, you can see QuickTest perform each step you recorded; a yellow arrow in the left margin of the Keyword View indicates the step that QuickTest is running.

## Analyzing Test Results

When QuickTest finishes running the test, the Test Results window opens.

Results Tree

Test Results Details

Iteration Summary

Status Summary

**Recording Results Summary**

**Test:** Recording  
**Results name:** Res1  
**Time Zone:** Central Standard Time  
**Run started:** 10/9/2005 - 8:42:26  
**Run ended:** 10/9/2005 - 8:42:44

Iteration #	Results
1	Done

Status	Times
Passed	0
Failed	0
Warnings	0

Result Details | Screen Recorder

For Help, press F1 | Ready

Initially, the Test Results window contains two panes for displaying the key elements of your test run.

- ▶ The left pane displays the results tree, an icon-based view of the steps that were performed while the test was running. The results tree is organized according to the Web pages visited during the test run and can be expanded (+) to view each step. The steps performed during the test run are represented by icons in the tree. You can instruct QuickTest to run a test or action more than once using different sets of data in each run. Each test run is called an iteration, and each iteration is numbered. (The test you ran had only one iteration.)
- ▶ The right pane displays the test results details. The iteration summary table indicates which iterations passed and which failed. The status summary table indicates the number of checkpoints or reports that passed, failed, and raised warnings during the test.

Your test run succeeded because QuickTest was able to navigate the Mercury Tours site just as the test was originally recorded. In this section, you will inspect the steps QuickTest performed when running your test, and how the application window appeared when a step was performed.

## 1 View the test results for a specific step.

In the results tree, expand (+) **Test Recording Summary** > **Recording Iteration 1 (Row 1)** > **Action1 Summary** > **Welcome Mercury Tours** > **Find a Flight**.

Highlight **fromPort.Select** in the results tree.

The screenshot displays the 'Recording [Res1] - Test Results' application window. On the left, the 'Results Tree' shows a hierarchy: Test Recording Summary > Run-Time Data Table > Recording Iteration 1 (Row 1) > Action1 Summary > Welcome: Mercury Tours > Find a Flight: Mercury > fromPort.Select (highlighted). Below this, the 'Test Results Details' pane shows a list of steps under 'Find a Flight: Mercury', with 'fromPort.Select' selected. The 'Still Image Capture Pane' at the bottom left is empty. The main area on the right shows a table with the following data:

Object	Details	Result	Time
fromPort.Select	"New York"	Done	08/01/2007 - 19:30:31

Below the table is a screenshot of a 'FLIGHT FINDER' web page. The page includes a description: 'Use our Flight Finder to search for the lowest fare on participating you've booked your flight, don't forget to visit the Mercury Tours H reserve lodging in your destination city.' The 'Flight Details' section shows: Type:  Round Trip  One Way; Passengers: 1; Departing From: New York (highlighted with a pink box); On: January 7; Arriving In: Acapulco; Returning: January 7.

The Test Results window now contains three panes, displaying:

- the results tree, with one step highlighted
- the test results details of the highlighted step
- the still image capture pane, showing a screen capture of the Web page on which the step was performed

When you click a page in the results tree, QuickTest displays the corresponding page in the application view. When you click a step (an operation performed on an object) in the results tree, the corresponding object is highlighted in the application view. In this case, the **Departing From** text box is highlighted.

**2 Close the Test Results window.**

Choose **File > Exit**.

You have now successfully created and run a test that reserves a flight from New York to San Francisco on the Mercury Tours Web site.



# 4

---

## Creating Checkpoints

In the previous lessons, you created and ran a test checking that a series of steps performed on the Mercury Tours Web site runs smoothly. A checkpoint verifies that expected information is displayed in your application while the test is running.

<b>In this lesson, you will learn about:</b>	<b>On page:</b>
Understanding Checkpoint Types	40
Checking Objects	42
Checking Pages	46
Checking Text	48
Checking Tables	50
Running and Analyzing a Test with Checkpoints	53

## Understanding Checkpoint Types

QuickTest Professional offers the following types of checkpoints:

Checkpoint Type	Description	Example of Use
Standard Checkpoint	Checks values of an object's properties.	Check that a radio button is selected.
Image Checkpoint	Checks the property values of an image.	Check that the image source file is correct.
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 Checkpoint	Checks that a text string is displayed in the appropriate place in a Web page or application.	Check whether the expected text string is displayed in the expected location in a test object.
Text Area Checkpoint	Checks that a text string is displayed within a defined area in a Windows-based application.	Check that an area of a dialog box includes text that was entered in another part of the application.
Bitmap Checkpoint	Checks an area of a Web page or application after capturing it as a bitmap.	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.

Checkpoint Type	Description	Example of Use
Accessibility Checkpoint	Identifies areas of a Web site to check for Section 508 compliancy.	Check if the images on a Web page include ALT properties, required by the W3C Web Content Accessibility Guidelines.
XML Checkpoint	Checks the data content of XML documents.	Check the content of an element to make sure that its tags, attributes, and values have not changed.  <b>Note:</b> 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.

You can add most checkpoints to your test either while recording or afterward. The following sections explain how to create some of the checkpoints described above, in the test you created in “Recording Tests” on page 23.

---

**Note:** When QuickTest creates a checkpoint, it assigns a name based on information inside the checkpoint—the checked value, for example. The checkpoint name remains unchanged, even if you subsequently modify the information on which it was based. Keep this in mind when looking for checkpoints displayed in the Keyword View. However, note that QuickTest may shorten the name displayed in the Keyword View.

---

For more information on how to create checkpoints, refer to the *QuickTest Professional User's Guide*.

## Checking Objects

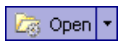
In this section, you will add a standard checkpoint in the Book a Flight page. This checkpoint verifies the value in the box containing the first name of the passenger.



### 1 Start QuickTest and open the Recording test.

If QuickTest is not already open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

► If the Welcome window opens, click **Open Existing**.



► If QuickTest opens without displaying the Welcome window, choose **File > Open > Test** or click the **Open** button down arrow and choose **Test**.

In the Open Test dialog box, locate and select the **Recording** test, then click **Open**.

### 2 Save the test as Checkpoint.

Choose **File > Save As**. Save the test as Checkpoint.

### 3 Confirm that the Active Screen option is enabled.



If you do not see the Active Screen at the bottom of the QuickTest window, click the **Active Screen** button, or choose **View > Active Screen**.

### 4 Locate the page where you want to add a standard checkpoint.

You want to add a checkpoint that checks the property values of the **First Name** edit box after the passenger's name has been entered in the box.

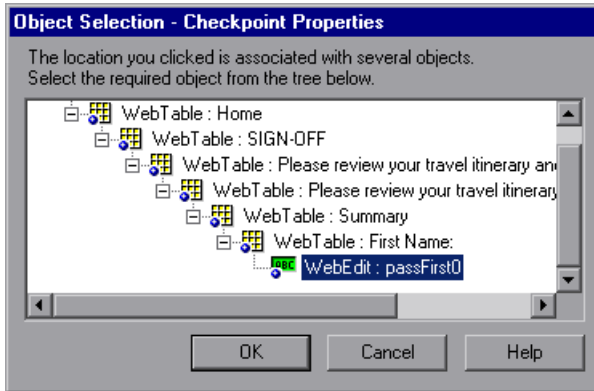
In the **Item** column of the Keyword View, expand **Action1 > Welcome: Mercury Tours**.



Highlight the **Book a Flight: Mercury** row. The Book a Flight page is displayed in the Active Screen and the object corresponding to the selected step is highlighted. You can see that the passenger's first name is displayed in the **First Name** box.

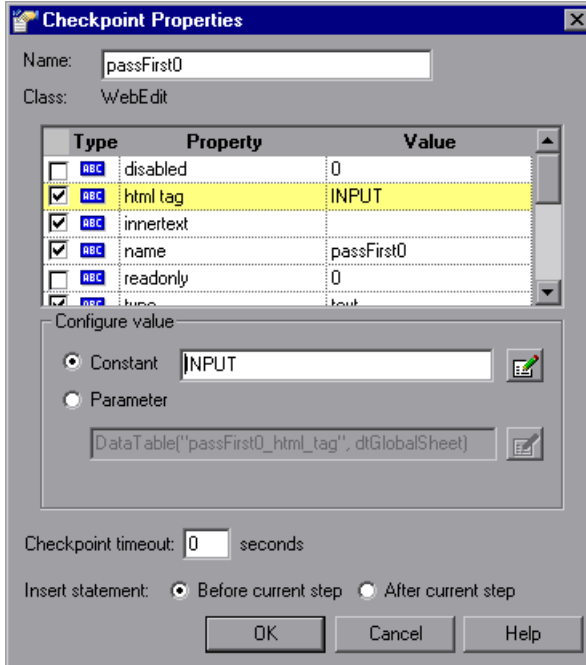
## 5 Create a standard checkpoint.

In the Active Screen, right-click the **First Name** box and choose **Insert Standard Checkpoint**. The Object Selection – Checkpoint Properties dialog box opens.



Confirm that **WebEdit: passFirst0** is highlighted, and click **OK**.

The Checkpoint Properties dialog box opens.



The dialog box displays the object's properties:

- ▶ The **Name** is the name of the object as defined in the HTML code of the Web page.
- ▶ The **Class** is the type of object. **WebEdit** indicates that the object is an edit box.
- ▶ The **ABC** icon in the **Type** column indicates that the value of the property is a constant.

For each object class, QuickTest recommends default property checks. The table below describes the default checks for the WebEdit class.

Property	Value	Explanation
html tag	INPUT	<b>INPUT</b> is the html tag as defined in the HTML source code.
innertext		In this case, the value of innertext is empty. The checkpoint checks that the value is empty.
name	passFirst0	<b>passFirst0</b> is the name of the edit box.
type	text	<b>text</b> is the type of object as defined in the HTML source code.
value	<FirstName> (The name you entered while recording)	The value entered in the edit box.

The passenger's first name is entered in the First Name box when the **passFirst0 Set...** step is performed. Therefore, in the **Insert statement** area of the Checkpoint Properties dialog box, select **After current step**. This inserts the checkpoint after the **passFirst0 Set...** step in which the first name is entered.

In the **Name** box, enter **CheckName** as the new checkpoint name.

Accept the rest of the settings as default and click **OK**. QuickTest adds a standard checkpoint step to your test below the selected step.

Book a Flight: Mercury				
passFirst0	Set	"Nicole"	Enter "Nicole" in the "passFirst0" edit box.	
passFirst0	Check	CheckPoint("CheckName")	Check whether the "passFirst0" edit box has the proper	
passLast0	Set	"Jones"	Enter "Jones" in the "passLast0" edit box.	

## 6 Save the test.



Choose **File > Save** or click the **Save** button.

You insert most checkpoints in the manner described above. Several special types of checkpoints are explained in the following sections.

## Checking Pages

In this section, you will add a page checkpoint to your test. The page checkpoint checks that the number of links and images in the page when you run your test is the same as when you recorded your test.

### 1 Locate the page where you want to add a page checkpoint.

In the **Item** column of the Keyword View, expand **Action1 > Welcome: Mercury Tours**.

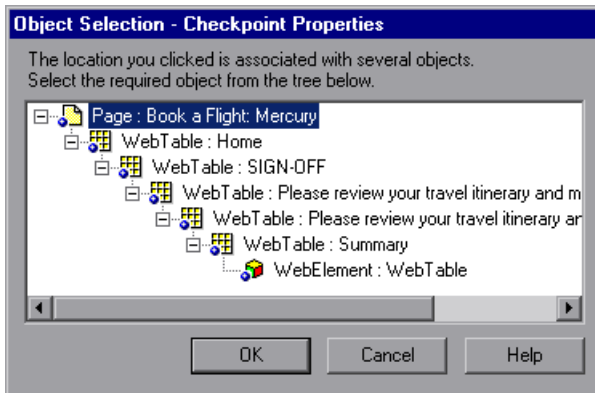


Highlight the **Book a Flight: Mercury** row. The Book a Flight page is displayed in the Active Screen.

### 2 Create a page checkpoint.

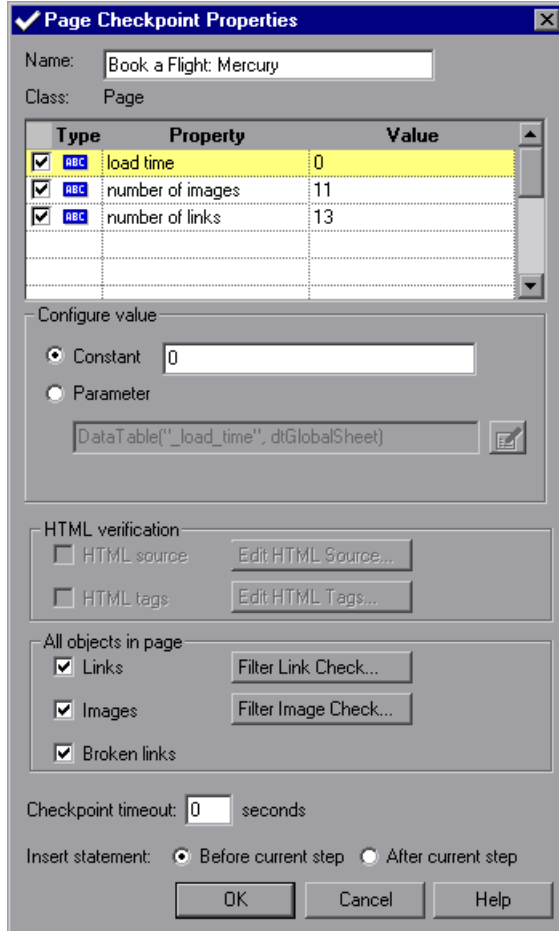
Right-click anywhere in the Active Screen, and choose **Insert Standard Checkpoint**. The Object Selection – Checkpoint Properties dialog box opens. This dialog box may include different elements, depending on where you click in the Active Screen.

Highlight **Page: Book a Flight: Mercury** (the top level) and click **OK**.





The Page Checkpoint Properties dialog box opens.



When you run the test, QuickTest checks the number of links and images in the page and the load time, as described in the pane at the top of the dialog box.

QuickTest also checks the actual target URL of each link and the actual source of each image.

In the **Name** box, enter **CheckLinks** as the new checkpoint name.

Accept the rest of the settings as default and click **OK**.

QuickTest adds the page checkpoint to your test. Because this is the first operation on the **Book a Flight: Mercury** page, the operation is added directly to the existing **Book a Flight: Mercury** page node. It is displayed in the Keyword View as a checkpoint operation on the **Book a Flight: Mercury** page.

A screenshot of the QuickTest Keyword View. The tree view shows a folder named 'Book a Flight: Mercury' containing a 'Check' operation. The 'Check' operation is expanded to show a 'CheckPoint("CheckLinks")' operation. The description of the operation is 'Check whether the "Book a Flight: Mercury" Web page'.

### 3 Save the test.



Choose **File > Save** or click the **Save** button.

## Checking Text

In this section, you will add a text checkpoint to your test to check whether **New York** is displayed in the **Flight Confirmation** page.

### 1 Locate the page where you want to add a text checkpoint.

In the Keyword View, expand **Action1 > Welcome: Mercury Tours**.

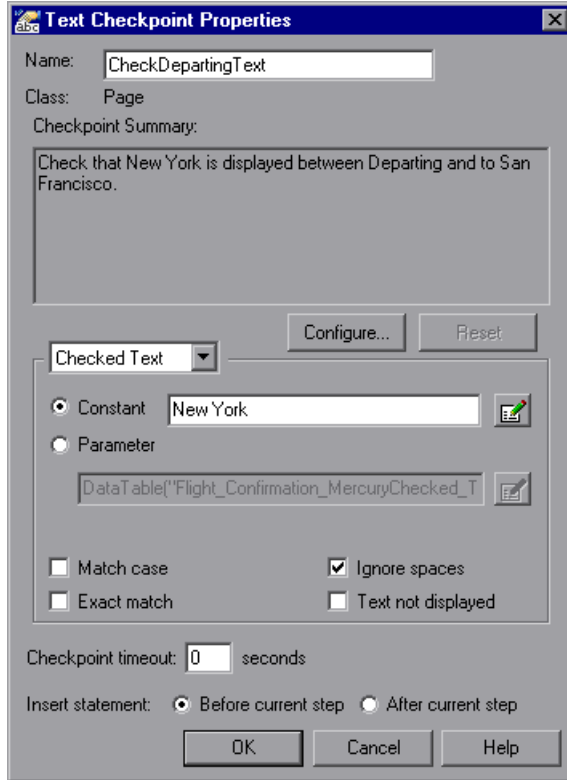
Highlight the **Flight Confirmation: Mercury** row. The page is displayed in the Active Screen.

### 2 Create a text checkpoint.

In the Active Screen, under **Departing**, highlight the text **New York** (include the space after it).

Right-click the highlighted text and choose **Insert Text Checkpoint**.

The Text Checkpoint Properties dialog box opens.



When **Checked Text** appears in the list box, the **Constant** field displays the text string you highlighted. This is the text QuickTest looks for when running the test.

In the **Name** box, enter **CheckDepartingText** as the new checkpoint name.

Accept the rest of the settings as default and click **OK**. QuickTest adds the text checkpoint to your test. It is displayed in the Keyword View as a checkpoint operation on the **Flight Confirmation: Mercury** page.

▼ Flight Confirmation: M... | Check | CheckPoint("Ne... | Check whether text in the "Flight Confirmation: Mercury" Web p

### 3 Save the test.



Choose **File > Save** or click the **Save** button.

## Checking Tables

In this section, you will add a table checkpoint to check the cost of the outbound flight, as displayed in the **Book a Flight: Mercury** page.

### 1 Locate the page where you want to add a table checkpoint.

In the Keyword View, expand **Welcome: Mercury Tours > Book a Flight: Mercury**.

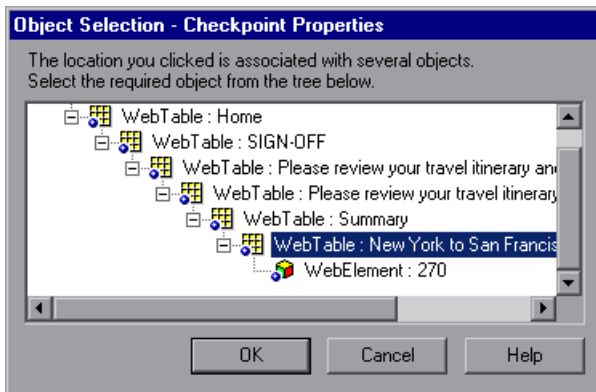
Highlight the **passFirst0** step (the step that enters the passenger's first name into the **First Name** edit box). The page is displayed in the Active Screen.

### 2 Create a table checkpoint.

In the Active Screen, right-click the price displayed for the first flight (New York to San Francisco)—**270**—and choose **Insert Standard Checkpoint**.

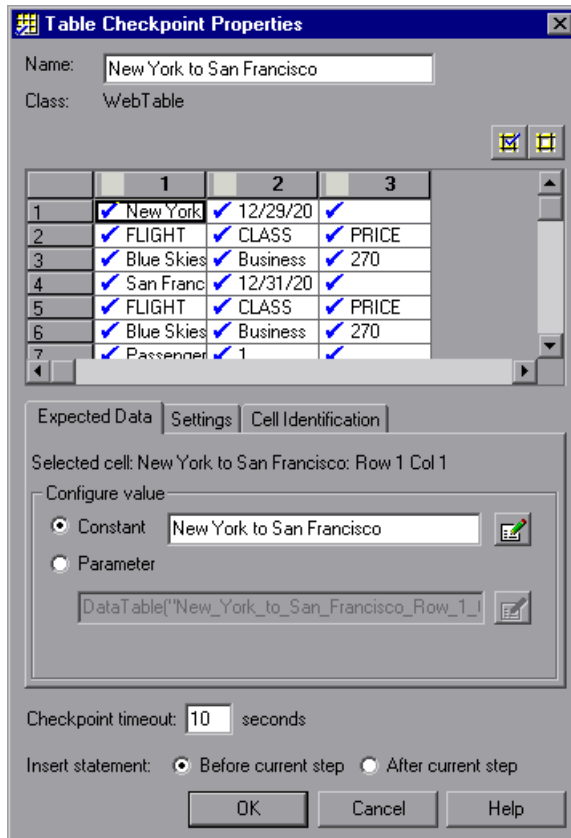
The Object Selection – Checkpoint Properties dialog box opens.

Select **WebTable: New York to San Francisco**.



Note that the table you selected is highlighted in the Active Screen. Click **OK**.

The Table Checkpoint Properties dialog box opens, showing the rows and columns of the table.



In the **Name** box, enter **CheckCost** as the new checkpoint name.

Note that by default, check marks appear in all cells. You can double-click a cell to toggle the cell selection, or double-click a row or column header to toggle the selection for all the cells in the selected row or column.

## Lesson 4 • Creating Checkpoints

Double-click each column header to clear the check marks. Double-click column **3**, row **3** to check the value of this cell. (QuickTest checks only cells containing check marks.)

	1	2	3
1	New York	12/29/20	
2	FLIGHT	CLASS	PRICE
3	Blue Skies	Coach	<input checked="" type="checkbox"/> 270
4	San Franc	12/31/20	
5	FLIGHT	CLASS	PRICE
6	Blue Skies	Coach	270
7	Passenger	1	



Scroll through the rows and columns to make sure that only the cell in column **3**, row **3** is checked. If any other cells are checked, select each checked cell and click the **Remove from Check** button.

---

**Tip:** You can change the column widths and row heights by dragging the boundaries of the column and row headers.

---

Accept the rest of the settings as default and click **OK**.

QuickTest adds the table checkpoint to your test. It is displayed in the Keyword View as a new step under the **Book a Flight: Mercury** page.

Book a Flight: Mercury	Check	Checkpoint("CheckLinks")	Check whether the "Book a Flight: Mercury" w
New York to San Francisco	Check	Checkpoint("CheckCost")	Check whether the content of specified cells in
passFirst0	Set	"Nicole"	Enter "Nicole" in the "passFirst0" edit box.

### 3 Save the test.



Choose **File > Save** or click the **Save** button.

## Running and Analyzing a Test with Checkpoints

In this section, you will review your test with checkpoints, run the test, and analyze the checkpoint results.

### 1 Expand the test and review your test.

Choose **View > Expand All** or select the **Action1** row and press the \* shortcut key on your number keypad. The Keyword View displayed is similar to the following:

Item	Operation	Value	Documentation
<b>Action1</b>			
Welcome: Mercury Tours			
Welcome: Mercury Tours			
userName	Set	"mercury"	Enter "mercury" in the "userName" edit box.
password	SetSecure	"4371dec7cf72594940a..."	Enter the encrypted string "4371dec7cf72594940a..."
Sign-In	Click	22,11	Click the "Sign-In" image.
Find a Flight: Mercury			
fromPort	Select	"New York"	Select the "New York" item from the "fromPort" list.
fromMonth	Select	"Dec"	Select the "Dec" item from the "fromMonth" list.
fromDay	Select	"29"	Select the "29" item from the "fromDay" list.
toPort	Select	"San Francisco"	Select the "San Francisco" item from the "toPort" list.
toMonth	Select	"Dec"	Select the "Dec" item from the "toMonth" list.
toDay	Select	"31"	Select the "31" item from the "toDay" list.
servClass	Select	"Business"	Select the "Business" radio button in the "servClass" list.
findFlights	Click	35,15	Click the "findFlights" image.
Select a Flight: Mercury			
reserveFlights	Click	36,10	Click the "reserveFlights" image.
Book a Flight: Mercury	Check	Checkpoint("CheckLinks")	Check whether the "Book a Flight: Mercury" Web page contains the "Book a Flight: Mercury" link.
New York to San Francisco	Check	Checkpoint("CheckCost")	Check whether the content of specified cells in the table is as expected.
passFirst0	Set	"Nicole"	Enter "Nicole" in the "passFirst0" edit box.
passFirst0	Check	Checkpoint("CheckName")	Check whether the "passFirst0" edit box has the expected text.
passLast0	Set	"Jones"	Enter "Jones" in the "passLast0" edit box.
creditnumber	Set	"12345"	Enter "12345" in the "creditnumber" edit box.
ticketLess	Set	"ON"	Set the state of the "ticketLess" check box to "ON".
buyFlights	Click	45,10	Click the "buyFlights" image.
Flight Confirmation: Mercury	Check	Checkpoint("CheckDeparture")	Check whether text in the "Flight Confirmation: Mercury" Web page is as expected.
home	Click		Click the "home" image.

Page Checkpoint

Table Checkpoint

Standard  
Checkpoint

Text Checkpoint

### 2 Start running your test.



Click **Run** or choose **Automation > Run**. The Run dialog box opens. Ensure that **New run results folder** is selected. Accept the default results folder name.

Click **OK**. When the test run is completed, the Test Results window opens.

### 3 View the test results.

When QuickTest finishes running the test, the Test Results window opens. The test result should be **Passed**, indicating that all checkpoints passed. If one or more checkpoints had failed, the test result would be **Failed**.

### 4 View the results of the page checkpoint.

In the results tree, expand (+) **Checkpoint Iteration 1 (Row 1) > Action1 Summary > Welcome: Mercury Tours > Book a Flight: Mercury**.

Highlight **Checkpoint "CheckLinks"**.

In the Details pane, you can review the details of the page checkpoint, which lists the items checked.

---

**Tip:** You can drag the bottom of the Details pane to enlarge it and/or scroll within the pane to view all of the checkpoint details.

---



The checkpoint passed because the actual values of the checked properties match the expected values.

The screenshot shows a software testing application window titled "checkpoint [Res1] - Test Results". The window is divided into several sections:

- Left Panel (Tree View):** Displays a hierarchical test structure. The "Action1 Summary" is expanded, showing a sequence of steps: "Welcome: Mercury Tours", "Find a Flight: Mercury", "Select a Flight: Mercury", "Book a Flight: Mercury", "Checkpoint 'CheckLinks'", "New York to San Francisco", "passFirst0.Set", "passLast0.Set", "creditnumber.Set", "ticketLess.Set", "buyFlights.Click", and "Flight Confirmation: Mercury". Each step is marked with a green checkmark, indicating it passed.
- Main Content Area:**
  - Standard Checkpoint "CheckLinks": Passed** (in green text).
  - Date and Time:** 09/01/2007 - 12:16:06
  - Details:**
    - Page Check:**

Property Name	Value
load time	
number of images	
number of links	
    - Checking the links captured addresses vs. the expected ones**
  - Image:** A screenshot of a Mercury Tours banner advertisement. The banner features the Mercury logo (a stylized plane), a group of people in a raft, and the text "one company" and "SIGN-OFF".
- Bottom Panel:** Contains tabs for "Result Details" and "Screen Recorder".

**5 View the results of the table checkpoint.**



In the results tree, **Book a Flight: Mercury** page, expand (+) **New York to San Francisco**.

Highlight **Checkpoint "CheckCost"**.

In the Details pane, you can review the details of the table checkpoint. You can also review the values of the table cells (cell values that were checked are displayed in black; cell values that were not checked are displayed in gray). The checkpoint passed because the actual value of the checked cell matches the expected value.

**Standard Checkpoint**  
**"CheckCost": Passed**

---

**Date and Time:** 11/10/2005 - 11:15:48

---

**Details**

Verification type: String Content. Settings: Exact match - ON; Ignore space - ON; Match case - OFF.  
 Results: Checked 1 cells; Succeeded: 1; Failed: 0

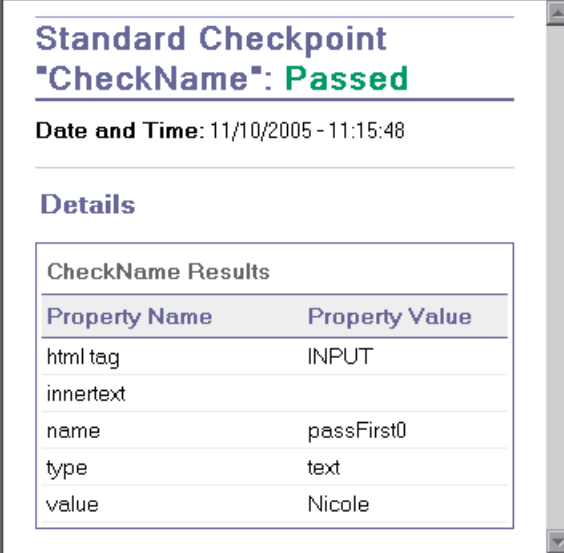
	1	2	3
1	New York	12/29/20	
2	FLIGHT	CLASS	PRICE
3	Blue Skies	Business	<b>270</b>
4	San Franc	12/31/20	
5	FLIGHT	CLASS	PRICE
6	Blue Skies	Business	270
7	Passenger	1	
8	Taxes:	\$44	
9	Total Pr	\$584	

**6 View the results of the standard checkpoint.**

In the results tree, **Book a Flight: Mercury** page, expand (+) **passFirst0**.

Highlight **Checkpoint "CheckName"**.

In the Details pane, you can review the details of the standard checkpoint, which lists the properties that were checked and their values. The checkpoint passed because the actual values of the checked properties match the expected values.



**Standard Checkpoint**  
**"CheckName": Passed**

**Date and Time:** 11/10/2005 - 11:15:48

**Details**

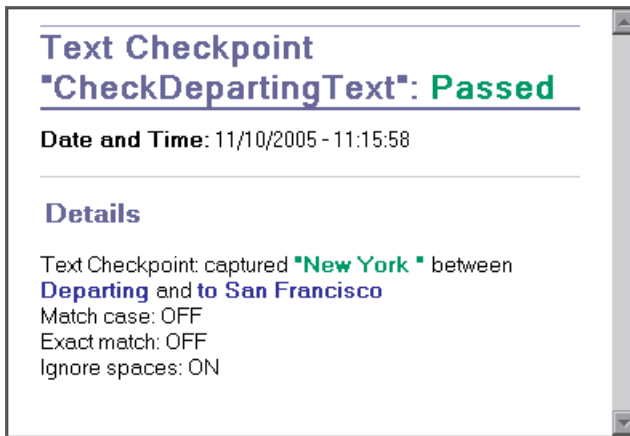
CheckName Results	
Property Name	Property Value
html tag	INPUT
innertext	
name	passFirst0
type	text
value	Nicole

**7 View the results of the text checkpoint.**

In the results tree, expand (+) **Checkpoint Iteration 1 (Row 1) > Action1 Summary > Welcome: Mercury Tours > Flight Confirmation: Mercury.**

Highlight **Checkpoint "CheckDepartingText"**.

In the Details pane, you can review the details of the text checkpoint. The checkpoint passed because the actual text matches the expected text.



**8 Close the Test Results window.**

Choose **File > Exit**.

# 5

---

## Parameterizing Tests

When you test your applications, you may want to check how the application performs the same operations with multiple sets of data. For example, suppose you want to check how your Web site responds to ten separate sets of data. You could record ten separate tests, each with its own set of data. Alternatively, you can create Data Table parameters so that your test runs ten times, each time using a different set of data.

<b>In this lesson, you will learn about:</b>	<b>On page:</b>
Defining a Data Table Parameter	60
Adding Parameter Values to a Data Table	62
Modifying Steps Affected by Parameterization	63
Running and Analyzing a Parameterized Test	65

## Defining a Data Table Parameter

In the previous lessons, you reserved a flight from New York to San Francisco. New York is a constant value, which means that New York is the departure city each time you run the test. In this exercise you will make the departure city a parameter so that you can use a different departure city for each test run.



### 1 Start QuickTest and open the Checkpoint test.

If QuickTest is not already open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

➤ If the Welcome window opens, click **Open Existing**.



➤ If QuickTest opens without displaying the Welcome window, choose **File > Open > Test** or click the **Open** button down arrow and choose **Test**.

In the Open Test dialog box, locate and select the **Checkpoint** test, then click **Open**.

### 2 Save the test as Parameter.

Choose **File > Save As**. Save the test as Parameter.

### 3 Confirm that the Active Screen option is enabled.



If you do not see the Active Screen at the bottom of the QuickTest window, click the **Active Screen** button, or choose **View > Active Screen**.


### 4 Confirm that the Data Table option is enabled.



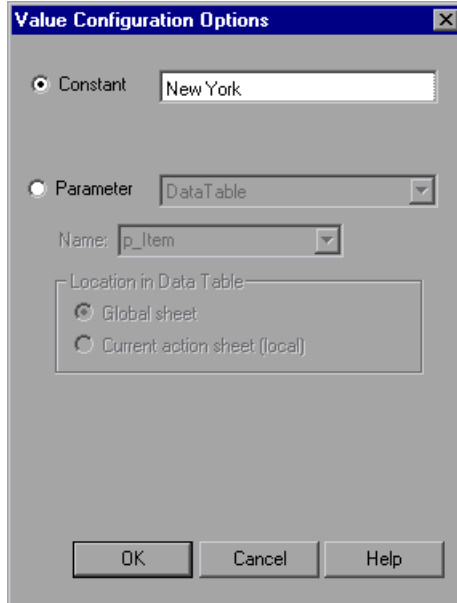
If you do not see the Data Table at the bottom of the QuickTest window, click the **Data Table** button, or choose **View > Data Table**.

### 5 Select the text to parameterize.

In the Keyword View, expand **Action1 > Welcome: Mercury Tours > Find a Flight: Mercury**.

In the **fromPort** row in the Keyword View, click the **Value** cell and then click the parameterization icon .

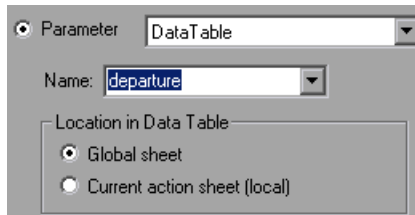
The Value Configuration Options dialog box opens.



## 6 Set the parameterization properties.

Select the **Parameter** radio button. This lets you replace the constant value (**New York**) with a parameter.

Confirm that the **DataTable** option is selected. This means that the value of the parameter will be taken from the QuickTest Data Table. The **Name** box is enabled and displays **p\_Item**. Delete this and enter **departure**.



Click **OK** to close the dialog box. QuickTest adds the **departure** parameter to the Data Table as a new column and inserts **New York** in the first row under it.

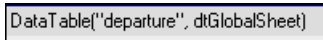
New York will be the first of several departure cities that QuickTest will use during test runs of the application.

The screenshot shows the QuickTest Keyword View. The 'Value' column for the 'Find a Flight: Mercury' step is highlighted, showing a Data Table parameter: `DataTable("departure", d...`. Below the keyword view, the 'Data Table' window is open, showing a table with columns labeled 'departure', 'depart\_flight\_price', 'C', 'D', and 'E'. The 'departure' column contains the value 'New York'. A red arrow points to the 'New York' cell with the text 'New Column in Data Table'. To the right, the 'Active Screen' window shows a flight search interface with buttons for 'Cruises', 'Destinations', and 'Vacations', and a 'Flight Details' section with a 'Passengers' dropdown set to '1' and a 'Departing From' dropdown set to 'New York'.

Note the change in the step's appearance in the Keyword View. Previously, the step was displayed as `fromPort Select New York`. Now, when you click the **Value** cell, the following information is displayed, indicating that the value is parameterized using a Data Table parameter called `departure`:



When you click any other cell the step is displayed as:



## Adding Parameter Values to a Data Table

As you saw, QuickTest displays parameter values in the Data Table. In this section, you will add two more departure cities to the Data Table, so that QuickTest can test the application with this data.

### 1 Enter additional cities in the departure column.

Click row **2** in the **departure** column and type Portland.

Click row **3** and type Seattle. Press **Enter**.

### 2 Save the test.



Choose **File > Save** or click the **Save** button.



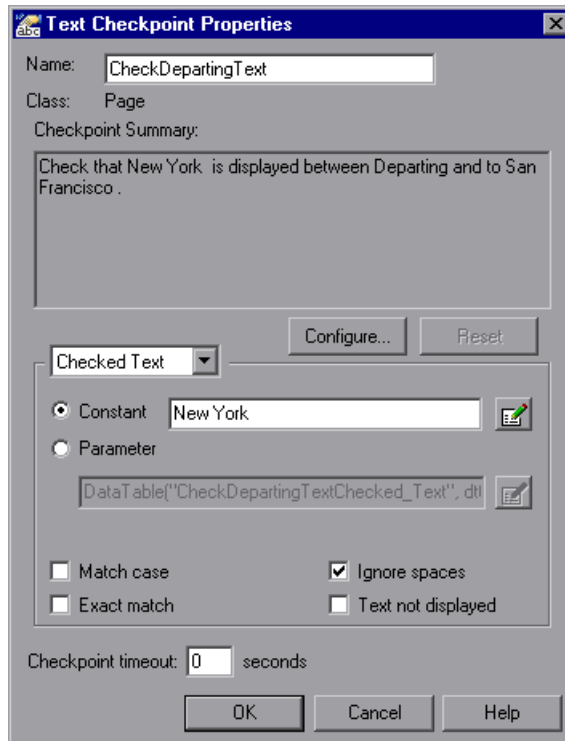
## Modifying Steps Affected by Parameterization

After parameterizing one step in a test, other test objects might be affected when the value of the parameterized step changes. If so, you must modify the expected values of those objects to match the value resulting from the parameterized step. In this section, you will modify the text checkpoint so that when running the test, QuickTest checks for the text that matches the current departure city.

### 1 Locate the text checkpoint to modify.

In the Keyword View, expand **Welcome: Mercury Tours**.

Right-click **Flight Confirmation: Mercury** and select **Checkpoint Properties**. The Text Checkpoint Properties dialog box opens.

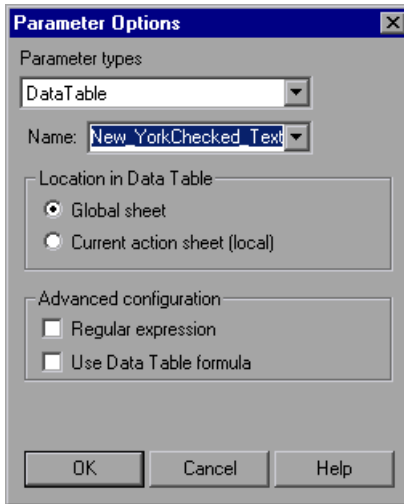


## 2 Parameterize the text checkpoint.

In the **Checked Text** area, **New York** is displayed in the **Constant** box. **New York** is the expected value for the checkpoint for every iteration.



Select **Parameter** and click the **Parameter Options** button. The Parameter Options dialog box opens.



In the **Name** box, select **departure**. This instructs the checkpoint to use the departure parameter values in the Data Table for the expected results.

Click **OK** to close the Parameter Options dialog box, and then click **OK** again to close the Text Checkpoint Properties dialog box. The checkpoint is now parameterized.

## 3 Save the test.



Choose **File > Save** or click the **Save** button.

## Running and Analyzing a Parameterized Test

You will now run the modified **Parameter** test. QuickTest will run the test three times, once for each departure city in the Data Table. Each test run is called an *iteration*.

### 1 Run the Parameter test.



Click **Run** on the Testing toolbar or choose **Automation > Run**. The Run dialog box opens.

Select **New run results folder** and accept the default results folder name.

Click **OK**. When the test run is completed, the Test Results window opens.

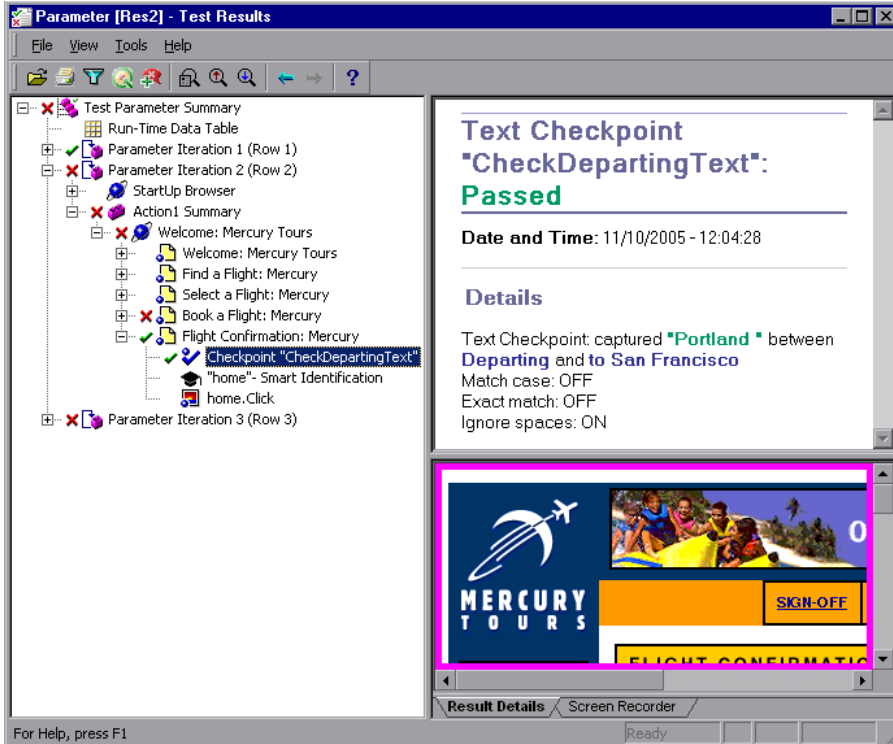
### 2 Examine the results summary.

The Test Results window shows that the second and third iterations of the test failed, even though the text checkpoint passed in all three iterations. See below for further information on why the iterations failed.

#### ► Iteration 2:

In the results tree, expand **Parameter Iteration 2 > Action1 Summary > Welcome Mercury Tours > Flight Confirmation: Mercury**.

## Highlight Checkpoint "CheckDepartingText".



The checkpoint results in the Details pane show that Portland was the expected result and the actual result. Therefore, the checkpoint passed. You can also see, in the Application pane, that the flight information reflects the flight from Portland.

### ► Iteration 3:

In the results tree, expand (+) **Parameter Iteration 3 > Action1 Summary > Welcome Mercury Tours > Flight Confirmation: Mercury.**

## Highlight Checkpoint "CheckDepartingText".

The checkpoint results in the Details pane show that Seattle was the expected result and the actual result. Therefore, the checkpoint passed. The Application pane shows the flight information for the flight from Seattle.

The table below summarizes the results of the text checkpoint in each iteration:

Iteration	Expected Text	Actual Text	Result
#1	New York	New York	Checkpoint Passed
#2	Portland	Portland	Checkpoint Passed
#3	Seattle	Seattle	Checkpoint Passed

Although the text checkpoints passed in all three iterations, the second and third iterations of the test still failed because the table checkpoint failed in these iterations—the ticket price changed when the departure city changed. In the next lesson, “Creating Output Values”, you will modify the table checkpoint so that when you run your test, QuickTest will update the expected ticket price in the checkpoint with the correct ticket price for the current departure city.

### 3 Close the Test Results window.

Choose **File > Exit** to close the Test Results window.



# 6

---

## Creating Output Values

In the previous lesson, you created parameters that inserted different data into each iteration of a test run. You can also retrieve data from your application and output it to the Data Table, using output values. This data can then be used at a later stage in the test. QuickTest displays the retrieved data, following the test run, in the Runtime Data Table.

For example, you can use an output value to verify that the date or flight number is correctly displayed in two different Web pages, by using the value obtained in one page as the expected text that QuickTest checks for in the other page.

In this lesson, you will learn about:	On page:
Creating an Output Value	69
Running and Analyzing a Test with Output Values	75

### Creating an Output Value

In the previous lesson, the second and third iterations of your test failed because the ticket price changed when the departure city changed. The checkpoint that checked the fare of the outbound flight in the **Book a Flight: Mercury** page did not update its expected value as the fare changed.

In this lesson, you will create an output value that retrieves the outbound fare from the **Select a Flight: Mercury** page in each test iteration. You will then modify the table checkpoint you created, so that it checks that the price displayed in the **Book a Flight: Mercury** page matches the price captured in the **Select a Flight: Mercury** page.



**1 Start QuickTest and open the Parameter test.**

If QuickTest is not already open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

► If the Welcome window opens, click **Open Existing**.



► If QuickTest opens without displaying the Welcome window, choose **File > Open** or click the **Open** button.

In the Open Test dialog box, locate and select the **Parameter** test, then click **Open**.

**2 Save the test as Output.**

Choose **File > Save As**. Save the test as Output.

**3 Confirm that the Active Screen option is enabled.**



If you do not see the Active Screen at the bottom of the QuickTest window, click the **Active Screen** button, or choose **View > Active Screen**.

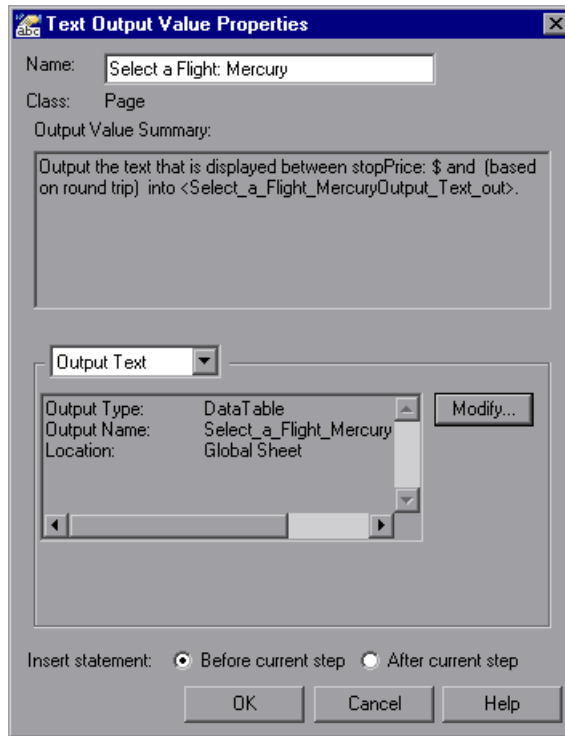
**4 Select the text you want to use as an output value.**

In the Keyword View, expand **Welcome: Mercury Tours** and click the **Select a Flight: Mercury** page. The page is displayed in the Active Screen.

In the Active Screen, highlight the fare, without the \$ sign, for the outbound flight (New York to San Francisco)—**270**.



Right-click the highlighted text and select **Insert Text Output**. The Text Output Value Properties dialog box opens.

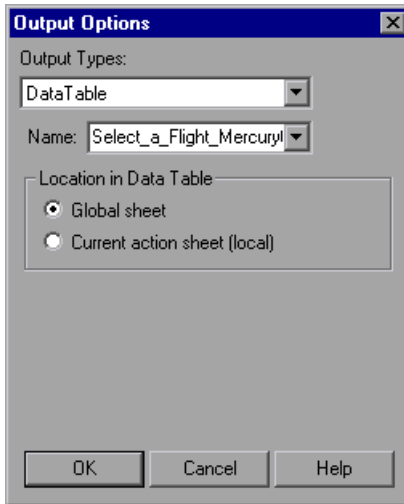


### 5 Rename the output value.

In the **Name** box, enter **PriceOutput** as the new output value name.

## 6 Set the output value settings.

Click the **Modify** button to open the Output Options dialog box.



The **Name** box displays **Select\_a\_Flight\_MercuryOutput\_Text\_out**. Delete this and enter **depart\_flight\_price**.

Click **OK** to accept the other default settings and close the Output Options dialog box. Click **OK** again to close the Text Output Value Properties dialog box. QuickTest adds the **depart\_flight\_price** column to the Data Table.

QuickTest adds the output value operation to the **Select a Flight: Mercury** page in the Keyword View.

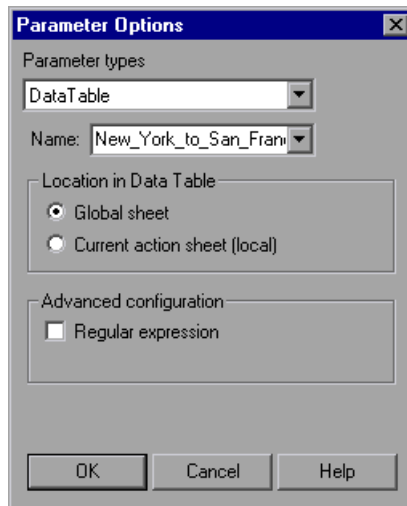
## 7 Modify the table checkpoint that checks the ticket price displayed in the flight Summary table.

In the Keyword View, expand **Welcome:Mercury Tours > Book a Flight: Mercury**.

Right-click **Checkpoint ("CheckCost")** in the **Value** column, and select **Checkpoint Properties**. The Table Checkpoint Properties dialog box opens.



Click column **3** in row **3** (the checked cell). In the Expected Data tab, select **Parameter** and click the **Parameter Options** button. The Parameter Options dialog box opens.

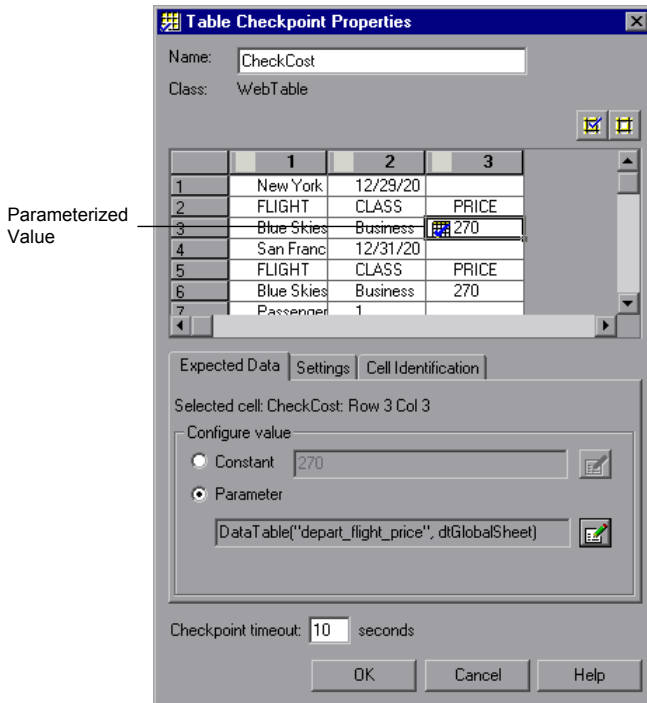


In the **Name** box, select **depart\_flight\_price**.

Click **OK** to close the Parameter Options dialog box.

## Lesson 6 • Creating Output Values

The Table Checkpoint Properties dialog box is redisplayed, with a parameterization icon next to the table value you parameterized.



Click **OK** to close the Table Checkpoint Properties dialog box. The checkpoint is now parameterized.

### 8 Save the test.



Choose **File > Save** or click the **Save** button.

## Running and Analyzing a Test with Output Values

You will now run the test and examine the results.

### 1 Run the Output test.



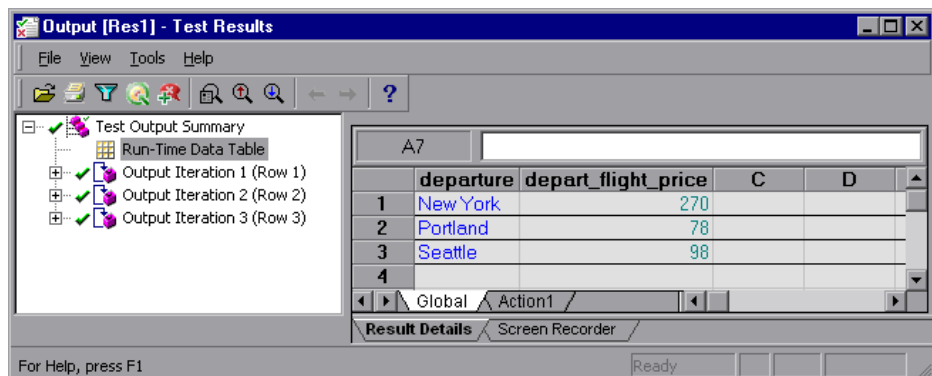
Click the **Run** button or choose **Automation > Run**. The Run dialog box opens.

Select **New run results folder** and accept the default results folder name.

Click **OK**. When the test run is completed, the Test Results window opens.

### 2 Examine the run-time data results.

In the Test Results window, select **Run-Time Data Table** from the results tree. The output values used during the test run are displayed in a grid. Note that a different price is shown in the **depart\_flight\_price** column for each iteration.



### 3 Examine the checkpoint results.

Choose **View > Expand All**.

In **Output Iteration 1 (Row 1)**, under the **Book a Flight: Mercury** page, click **Checkpoint "CheckCost"**.

Note that in the Details pane, the checkpoint is marked in green as passed because the price of the ticket for the first flight in the Summary table matched the expected value obtained from the Select a Flight page.

Examine the checkpoints for Iterations 2 and 3 as well. In the Details pane, these checkpoints are also marked in green as passed, for the same reason.

**4 Close the Test Results window.**

Choose **File > Exit** to close the Test Results window.

# 7

---

## Using Regular Expressions

In Lesson 4, “Creating Checkpoints,” you created a text checkpoint that searched for a specific text string. You can use regular expressions to increase the flexibility and adaptability of your tests.

In this lesson, you will learn about:	On page:
Regular Expression Syntax	77
Working with Regular Expressions	78
Running and Analyzing a Test with Regular Expressions	81

### Regular Expression Syntax

Regular expressions enable QuickTest to identify objects and text strings with varying values.

A regular expression is a string that specifies a complex search phrase. By using special characters such as a period (.), asterisk (\*), caret (^), and brackets ([ ]), you define the conditions of the search. For more information on regular expression syntax, refer to the *QuickTest Professional User's Guide*.

## Working with Regular Expressions

In this lesson, you will create a text checkpoint on a date text string that changes according to the selected flight date. You can define the date as a regular expression so that the checkpoint checks that the captured text string matches the expected format, rather than checking the exact text.

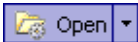
To do this, you will create a text checkpoint with a regular expression that will match any single character within a defined range.



### 1 Start QuickTest and open the Recording test.

If QuickTest is not already open, choose **Start > Programs > QuickTest Professional > QuickTest Professional**.

► If the Welcome window opens, click **Open Existing**.



► If QuickTest opens without displaying the Welcome window, choose **File > Open** or click the **Open** button.

In the Open Test dialog box, locate and select the **Recording** test, then click **Open**.

### 2 Save the test as RegExpression.

Choose **File > Save As**. Save the test as RegExpression.

### 3 Confirm that the Active Screen option is enabled.



If you do not see the Active Screen at the bottom of the QuickTest window, click the **Active Screen** button, or choose **View > Active Screen**.

### 4 Select the text for which you will create the checkpoint.

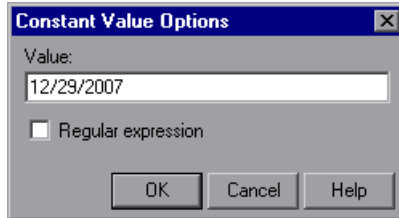
In the Keyword View, expand **Welcome: Mercury: Tours** and click the **Select a Flight: Mercury** page. The page is displayed in the Active Screen.

In the Active Screen, scroll up and highlight the date for the outbound flight, New York to San Francisco (12/29/2007). Right-click the highlighted string and select **Insert Text Checkpoint**. The Text Checkpoint Properties dialog box opens.



**5 Enter the regular expression for the text you want to check.**

In the **Checked Text** area, click the **Constant Value Options** button. The Constant Value Options dialog box opens.



In the **Value** box, replace the displayed date with `[0-1][0-9]/[0-3][0-9]/200[0-9]`. This instructs QuickTest to check that each character in the selected text matches the number-range format defined by the regular expression. The expression checks for the following format: MM/DD/200Y.

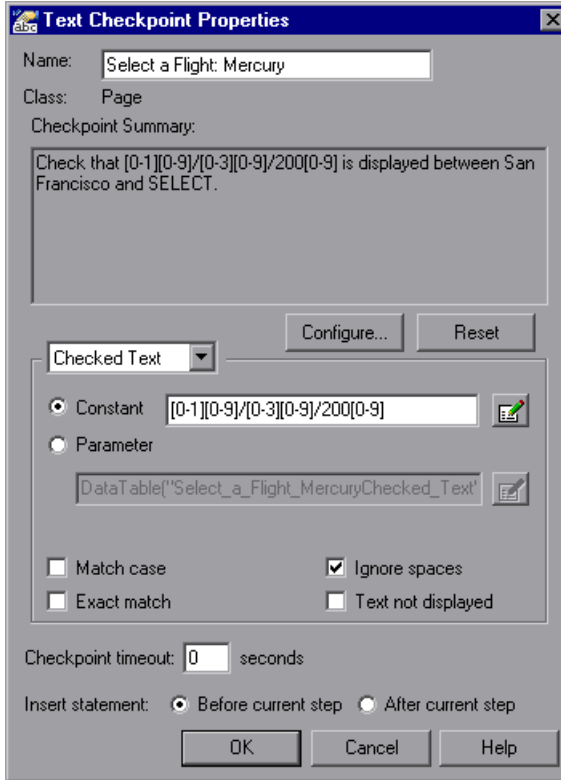
Select the **Regular expression** check box. A message box prompts you to insert the backslash character before each special character. Click **No**.

---

**Note:** Clicking **Yes** would cause QuickTest to treat the special characters ( `[` ), ( `-` ) and ( `]` ) as literal characters and not as a regular expression.

---

Click **OK** to close the Constant Value Options dialog box.



## 6 Rename the checkpoint.

In the **Name** box, enter **CheckExpectedText** as the new checkpoint name.

Click **OK** to accept the other default settings and close the Text Checkpoint Properties dialog box.

QuickTest adds the text checkpoint to your test. It is displayed as a check operation on the **Select a Flight: Mercury** page in the Keyword View.

## 7 Save the test.



Choose **File > Save** or click the **Save** button.

## Running and Analyzing a Test with Regular Expressions

In this exercise you will run the test and examine the checkpoint results.

### 1 Run the RegExpression test.



Click the **Run** button or choose **Automation > Run**. The Run dialog box opens.

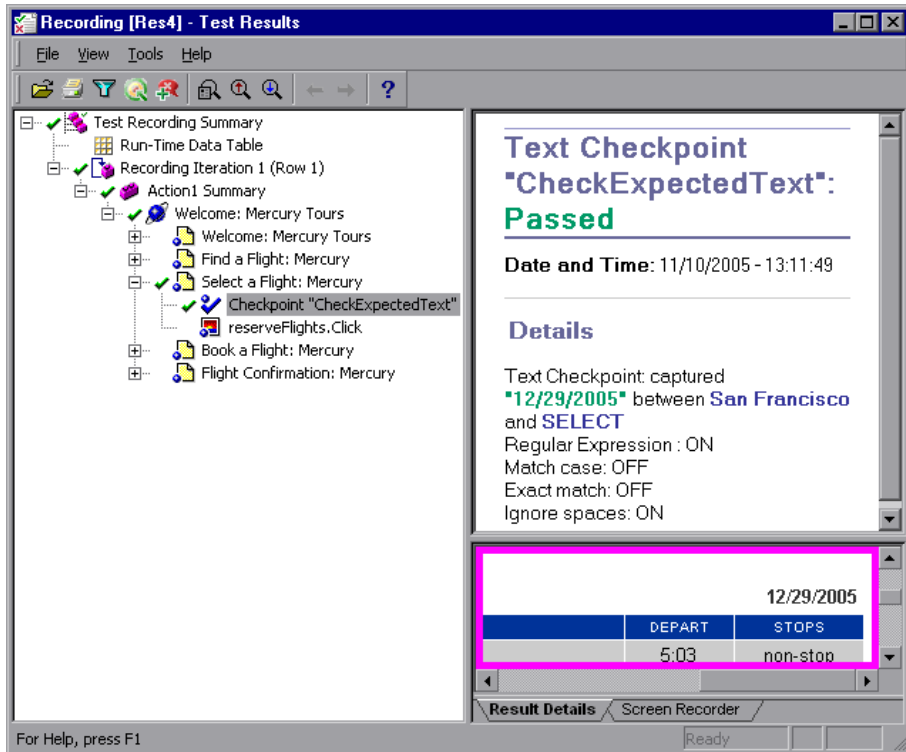
Select **New run results folder** and accept the default results folder name.

Click **OK**. When the test run is completed, the Test Results window opens.

### 2 Examine the checkpoint results.

In the results tree, expand (+) **Test RegExpression Summary > RegExpression Iteration 1 (Row 1) > Action1 Summary > Welcome: Mercury Tours > Select a Flight: Mercury**

Select **Checkpoint "CheckExpectedText"**.



The checkpoint passed because the text was displayed in the format specified by the regular expression.

**3 Exit the Test Results window.**

Choose **File > Exit** to close the Test Results window.

# 8

---

## Dividing Tests into Multiple Actions

Actions are test units that divide your test into logical sections. When you create a new test, it contains a call to one action. By dividing your tests into calls to multiple actions, you can design more modular and efficient tests.

In this lesson, you will learn about:	On page:
Working with Multiple Actions	83
Creating New Actions	84
Inserting Existing Actions	93
Parameterizing an Action	99
Running and Analyzing a Multi-Action Test	103

### Working with Multiple Actions

If you examine one of the tests you created in the previous lessons, you will see that it can be divided into several distinct processes:

- You logged into the Mercury Tours site.
- You submitted a flight order.
- You logged out.

Assume that you wanted to run your test for three different flight orders. As we saw in Lesson 5, “Parameterizing Tests,” you can parameterize your test so that it runs the test three times using three different sets of data. You can also organize your test so that only the second procedure runs three times, simulating a single user logging in, ordering three flights, and logging out. You do this by dividing your test into calls to different actions.

## Lesson 8 • Dividing Tests into Multiple Actions

To divide your test into calls to different actions, you can insert a call to a new action, split an existing action into calls to two actions, insert a call to a copy of an existing action, or insert a call to an existing action.

You can insert calls to actions into your test during your recording session or afterward. Use one of the following menu options or toolbar buttons to add actions to your test:



- ▶ **Insert > Call to New Action** or use the **Insert Call to New Action** button. You can click the down arrow and choose **Call to Copy of Action** or **Call to Existing Action**.



- ▶ **Edit > Action > Split Action** or use the **Split Action** button.
- ▶ **Insert > Call to Copy of Action** or right-click an action and choose **Insert Call to Copy of Action**.
- ▶ **Insert > Call to Existing Action** or right-click an action and choose **Insert Call to Existing Action**.

## Creating New Actions

In this exercise you will create a test and divide it into action calls.

### Recording the Test

In this exercise you will create a test and divide it into action calls.



#### 1 Start QuickTest and open a new test.

For more information, see step 1 of “Recording a Test” on page 24.

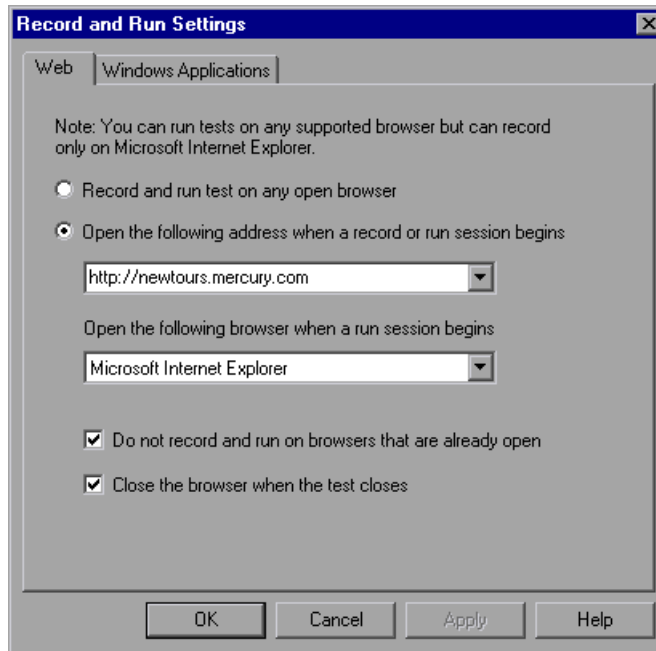
#### 2 Start recording on the Mercury Tours Web site.

In the coming steps, you will record a test similar to the one recorded in the “Recording Tests” lesson. However, this test is designed slightly differently to optimize the test for a multi-action test.

Confirm that all Web browsers are closed.



Choose **Automation > Record** or click the **Record** button. The Record and Run Settings dialog box opens.



- In the Web tab, select **Open the following address when a record or run session begins**.

Confirm that the URL in the first box is <http://newtours.mercury.com> and in the second box, choose a browser on which to run your test session.

Confirm that the **Do not record and run on browsers that are already open** and **Close the browser when the test closes** options are selected.

- In the Windows Applications tab, confirm that **Record and run only on** is selected, and that all three check boxes are cleared.

Click **OK**.

QuickTest begins recording, and your browser opens to the Mercury Tours Web site.

### 3 Sign in to the Mercury Tours Web site.

In the **User Name** and **Password** boxes, type the name and password you registered with Mercury Tours.

Click **Sign-In**.

The Flight Finder page opens.

### 4 Enter flight details.

Make the following selections:

- Departing From: **New York**
- On: **December 29**
- Arriving In: **San Francisco**
- Returning: **December 31**

---

**Note:** For the purposes of this tutorial, you must select each of the above options, even if one or more options are already selected.

---

Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

### 5 Select a flight.

Click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

### 6 Enter required passenger and purchase information.

Enter the required information (fields with red text labels) in the Passengers and Credit Card sections.

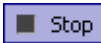
At the bottom of the page, click **SECURE PURCHASE**. The Flight Confirmation page opens.



**7 Return the browser to the Flight Finder page.**

At the bottom of the Flight Confirmation page, click **BACK TO FLIGHTS** button. The Flight Finder page opens.

Ending the recording on the Flight Finder page enables you to create an action that starts and ends on the same page, so that you can call the action multiple times if you wish.

**8 Stop Recording.**

In QuickTest, click **Stop** to stop the recording process.

**9 Save the test as ActionA.**

Choose **File > Save**. Save the test as ActionA.

**Dividing the Test into Two Actions**

Since your test currently contains two main processes: signing-in and purchasing a flight, it makes sense to divide the test into separate actions.

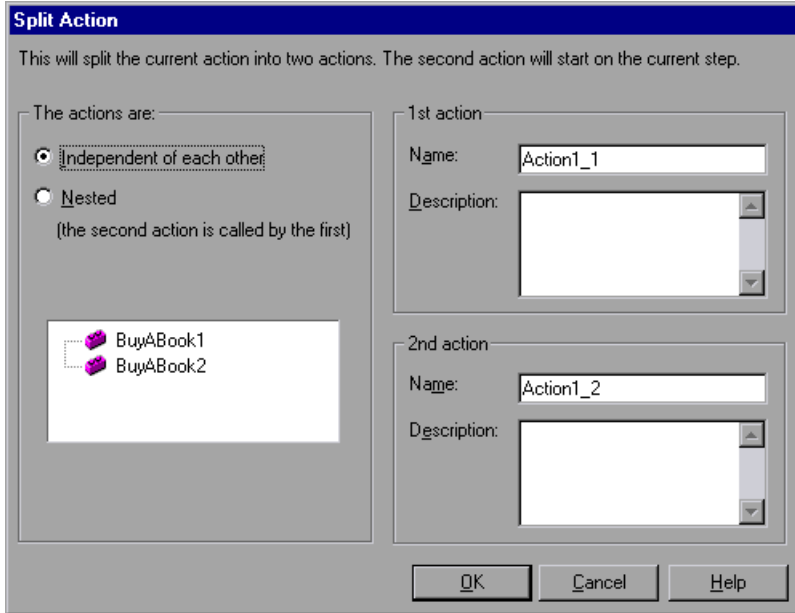
**1 Select the page where you want the second action to begin.**

In the Keyword View, expand **Action1 > Welcome: Mercury Tours**.

**2 Split the test into two actions.**

Select the first **Find a Flight: Mercury** page and choose **Edit > Action > Split Action** or click the **Split Action** button. This splits the current action into two actions. The new action will start on the **Find a Flight: Mercury** page, which is the currently selected step.

The Split Action dialog box opens.



**3 Enter names and descriptions for the two actions.**

In the **Name** box of the **1st action** type: Sign\_in.

In the **Description** box of the **1st action**, type: Enters user name and password to sign in.

In the **Name** box of the **2nd action** type: FlightOrder.

In the **Description** box of the **2nd action**, type: Chooses and purchases a flight.

Click **OK** to close the Split Action dialog box and accept the other default selections. The two actions are displayed in the Keyword View.

Choose **View > Expand All** to examine the contents of each action. Note that the **FlightOrder** action begins and ends with the **Find a Flight: Mercury** page.

## Inserting a Call to a New Action

Now you will insert a call to a new action and record the step of returning to the Mercury Tours home page so that the entire test begins and ends on the same page.

### 1 Insert the ReturnHome action.



Choose **Insert > Call to New Action** or click the **Insert Call to New Action** button. The Insert Call to New Action dialog box opens.

**Insert Call to New Action**

Name :

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

Enter **ReturnHome** in the **Name** box, confirm that **At the end of the test** is selected, and click **OK**. The dialog box closes and the **ReturnHome** action is added to the end of your test.

### 2 Prepare the application for recording.

Confirm that your Web browser is open to the Flight Finder page.

If your browser is not open, open it to the Mercury Tours site (<http://newtours.mercury.com>) and sign in. The Flight Finder page opens.

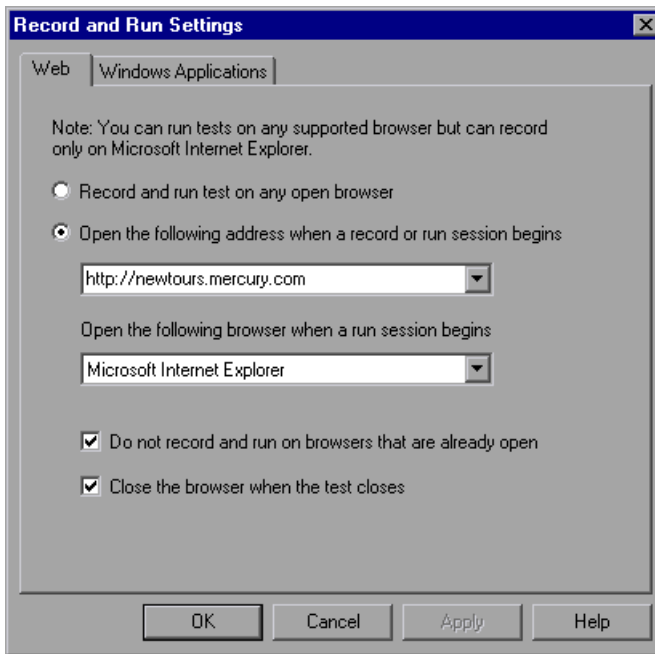
Position the applications on your desktop so that you can see both QuickTest and the Web browser.

### 3 Change the Record and Run settings.

Currently, your Record and Run settings are set so that a new browser opens each time you begin recording your test. However, because you want to record a step on a browser that you have already opened and prepared, you need to change the Record and Run settings before you begin recording.

In QuickTest, choose **Automation > Record and Run Settings**.

The Record and Run Settings dialog box opens.



In the **Web** tab, select **Record and run test on any open browser** and click **OK**.

#### 4 Record the ReturnHome action.

In the Keyword View, select the new **ReturnHome** action icon.



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

Click the **Home** navigation link on the left side of the Flight Finder page. The Mercury Tours home page opens.



In QuickTest, click **Stop** to stop the recording process.

#### 5 Return the Record and Run Settings to their original settings.

You changed the record and run settings for the purposes of recording the **ReturnHome** action, but this change also affects the way QuickTest runs the test. You want QuickTest to open a new browser to the Mercury Tours home page when you run the test, so you need to reapply the original settings.

Choose **Automation > Record and Run Settings**. The Record and Run Settings dialog box opens.

In the Web tab, select **Open the following address when a record or run session begins**. Confirm that <http://newtours.mercury.com> is displayed in the first box and click **OK**.

#### 6 Save the test.




Choose **File > Save** or click the **Save** button.

### Setting Actions as Reusable Actions

Reusable actions are like functions in programming. When you have a process that you may need to include in several tests, you record, modify, and enhance the steps of the process and save them in a reusable action. Then you can call the action from other tests, rather than recording, modifying, and enhancing the same steps each time.

To call an action from another test, you must make it a reusable action in the test with which it was stored. The **Sign\_in** and **ReturnHome** action you created in this lesson can apply to almost any test you might create on the Mercury Tours site, so it makes sense to make them reusable actions.

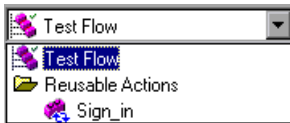
### 1 Make the Sign\_in action reusable.

In the Keyword View, right-click the **Sign\_in** action and choose **Action Properties**. The Action Properties dialog box opens. Select the **Reusable action** check box and click **OK**. If a message box is displayed, read the message, click **Do not show this dialog box again**, and click **OK**. The action icon changes to a reusable action icon .

### 2 View the details of the Sign\_in action.

Note that you no longer have the option to expand the **Sign\_in** action within the Keyword View.

If the Action List was not previously displayed above the Keyword View, it is displayed after you convert the action to a reusable action. The Action List displays a list of all reusable and external actions in the test.



View the details of the **Sign\_in** action by selecting **Sign\_in** from the Action List. The details of the **Sign\_in** action are displayed.

In the Action List, select **Test Flow** to return to the main flow of the test in the Keyword View.

### 3 Make the ReturnHome action reusable.

Repeat steps 1 and 2 for the **ReturnHome** action.

### 4 Save the test.



Choose **File > Save** or click the **Save** button.

## Inserting Existing Actions

When you plan a suite of tests, you may realize that each test requires one or more identical activities, such as signing in. Once you have created the action and stored it with one test, you can insert either a call to a copy of the existing action, or a call to the existing action, into other tests.

When you insert a call to a copy of an existing action, you can make changes to the copied action, and your changes will neither affect, nor be affected by, any other test. Calls to existing actions, however, are read-only in the calling test. They can be modified only in the test in which they were stored. Calls to existing actions enable you to call the same action from several tests and make it easy to maintain tests, because when your application changes you only have to update the existing action stored with the original test.

In the following exercises you will create a new test that is similar to the **ActionA** test, except that the **Sign\_in** and **ReturnHome** actions are external actions (calls to existing actions stored with other tests) and the **FlightOrder** action is slightly modified.

### Inserting Calls to Actions

First, you will insert calls to the reusable **Sign\_in** and **ReturnHome** actions from **ActionA** into **ActionB**.



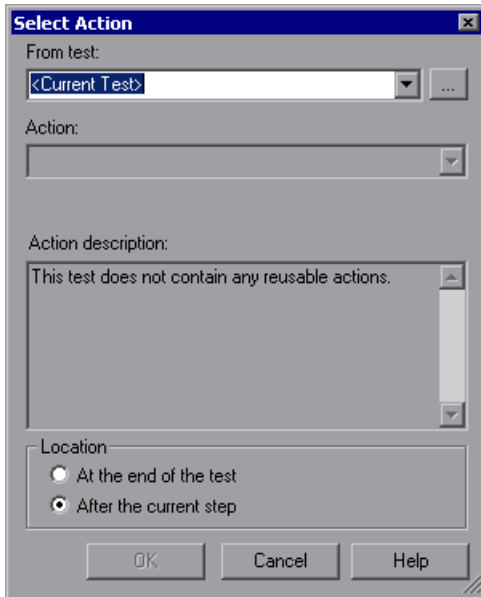
#### 1 Open a new test.

For more information, see step 1 of “Recording a Test” on page 24.


## 2 Insert a call to the Sign\_in action.

Choose **Insert > Call to Existing Action**.

The Select Action dialog box opens.



Click the browse button next to the **From test** box. Select the **ActionA** test in the Open Test dialog box, and click **Open**. Note that only reusable actions from the **ActionA** test are displayed in the Select Action dialog box.

Select the **Sign\_in** action. In the **Location** area, select **At the end of the test**. Click **OK**. The call to **Sign\_in [ActionA]** action is added to the end of the test as a call to an external action with the  icon.

To view the steps in the **Sign\_in** action, double-click it or select it from the Action List. The action call you inserted is read-only. To modify the action, you must open the original (**ActionA**) test and modify the action there.

Select **Test Flow** in the Action List to return to the main flow of the test in the Keyword View.



### 3 Insert a call to the ReturnHome action.

Use the instructions in step 2 to insert a call to the **ReturnHome** action from **ActionA**. The **ReturnHome** action is added to the end of your test.

---

**Note:** When you insert a call to the **ReturnHome** action, **Sign\_in [ActionA]** is selected in the Keyword View. Since the **Sign\_in** action is read-only, you cannot select the location in your test in which to insert the call to the **ReturnHome** action. Therefore, the action call is automatically added at the end of the test.

---

### 4 Save the test.



Choose **File > Save** or click the **Save** button. Save the test as ActionB.

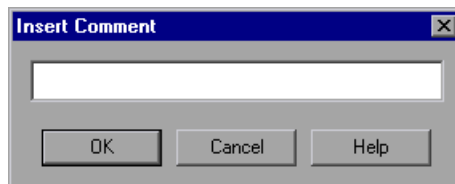
### 5 Modify the original ReturnHome action.

Choose **File > Open > Test**. Browse to **ActionA**, and click **Open**.

Double-click the **ReturnHome** action icon to display its contents. (This is equivalent to selecting the **ReturnHome** action from the Action List.)

Make sure that the **Comment** column is displayed in the Keyword View. If it is not, right-click on any Keyword View column header and select **Comment** from the displayed list.

In the Keyword View, select the **ReturnHome** action icon and choose **Insert > Comment** to add a comment about the purpose of the **Home** step. The Insert Comment dialog box opens.



In the edit box, enter: Return to page used in first step of the test. Click **OK**.

The comment step is added as the first step in the action, and you can see the comment text in the **Comment** column of that step.

## Lesson 8 • Dividing Tests into Multiple Actions

If the comment is too long to see, it is displayed in a tooltip when you move your cursor over the **Comment** column.

Item	Operation	Comment	Documentation
ReturnHome			
Comment		Return to page used ...	Return to page used in first step of the test
Find a Flight: Mercury			
Find a Flight: Mercury			
Home	Click		Click the "Home" link.
Welcome: Mercury Tours	Sync		Wait for the Web page to synchronize before ...

### 6 Save the test.



Choose **File > Save** or click the **Save** button.

### 7 Confirm that the modification made in the original ReturnHome action is also included in the calling test.

Choose **File > Open > Test**. Browse to **ActionB** and click **Open**.

Double-click the **ReturnHome [ActionA]** action icon to display it. Note that the comment you added to the original action (**ActionA**) also exists in the calling test (**ActionB**).

In the Action toolbar, select **Test Flow** from the Action List to return to the main flow of the test in the Keyword View.

## Inserting a Copy of an Action

Now, you will insert a copy of the **FlightOrder** action.

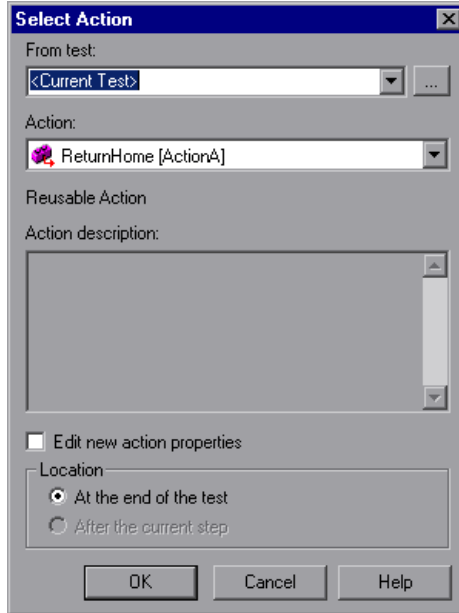
### 1 Confirm that the ActionB test is open.

If not, choose **File > Open > Test**. Browse to **ActionB** and click **Open**.

### 2 Insert a copy of the FlightOrder action.

Choose **Insert > Call to Copy of Action**.

The Select Action dialog box opens.



Click the browse button next to the **From test** box. Select the **ActionA** test in the Open Test dialog box, and click **Open**. The actions from the **ActionA** test are displayed in the Action List.

Select the **FlightOrder** action. Note that the description of the selected action is displayed in the **Action description** box.

Click **OK**. The **Copy of FlightOrder** action is added to the end of the test. Note that the call to the copy of the action you inserted is just like any other action call in your test. You can modify the steps in the call to the copied action just as you would those of any other action call.

### 3 Modify the Copy of FlightOrder action.

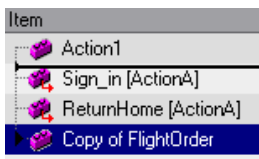
In the Keyword View, expand **Copy of FlightOrder > Welcome: Mercury Tours > Find a Flight: Mercury**. Select **fromPort** and click the **Value** cell.

Change the value in the **Value** cell to "London" (including the quotation marks) and click any other row in the Keyword View.

#### 4 Call the Copy of FlightOrder action after the Sign\_in action.

You want the **Copy of FlightOrder** action to be immediately after the **Sign\_in** action. Since you cannot drag any action to be immediately after a reusable action, you must perform the following steps to change the order of the action calls.

Select the **Copy of FlightOrder** action and drag it immediately below the **Action1** action in the Keyword View. A black line indicates the new position. To ensure that you position the **Copy of FlightOrder** action as a sibling of **Action1** and not as a child, drag the line to the left until it meets the vertical dotted line.



Then, select the **Sign\_in** action and drag it immediately below the **Action1** action in the Keyword View. Ensure that you position the **Sign\_in** action as a sibling of **Action1** and not as a child. The **Copy of FlightOrder** action is now immediately after the **Sign\_in** action.

#### 5 Delete Action1 from the test.

Every new test contains a default **Action1** action call. You can now delete the **Action1** action call.

In the Keyword View, select **Action1** and choose **Edit > Delete** or press the **Delete** key on your keyboard. A confirmation message opens. Click **Yes** to delete the action call.

#### 6 Save the test.



Choose **File > Save** or click the **Save** button.

## 7 Confirm that the original FlightOrder action was not modified.

Choose **File > Open > Test**. Browse to **ActionA**, and click **Open**.

Expand **FlightOrder > Welcome: Mercury Tours > Find a Flight: Mercury**.

Note that the change of departure city you made in the **ActionB** test did not affect the original action in **ActionA** (New York is still the departure city in **ActionA**).

## Parameterizing an Action

Open the **ActionB** test. If you look at the Data Table at the bottom of the QuickTest window in the **ActionB** test, you will see four tabs: **Global**, **Sign\_in [ActionA]**, **ReturnHome [ActionA]**, and **Copy of FlightOrder**.



**Note:** If the Data Table is not displayed, choose **View > Data Table** to display it, or click the **Data Table** button.

---

The **Global** tab is a data sheet whose data is used for the entire test. If five rows of data are displayed in the **Global** table, the test will run five times. In addition, you can create data sets for each action, using the relevant action sheet. If you parameterize a step using an action parameter and enter five rows of data in the corresponding sheet, you can define that action to run five times within each test iteration.

---

**Note:** The **Sign\_in [ActionA]** and **ReturnHome [ActionA]** data sheets are displayed in gray and cannot be edited because each of these data sheets belong to the corresponding called action and can be edited from the called action's original test only.


---

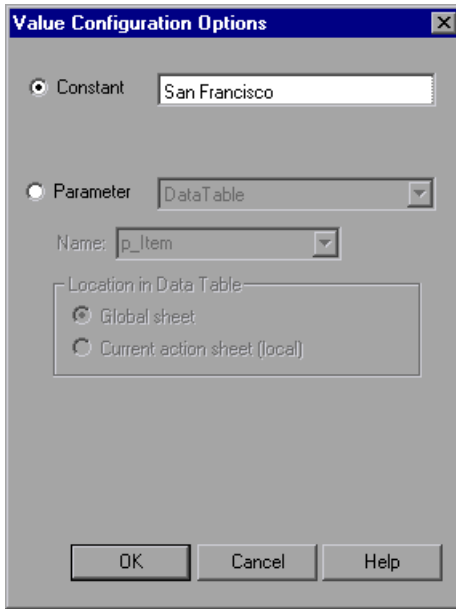
## Lesson 8 • Dividing Tests into Multiple Actions

In this section, you will create a data set for the **FlightOrder** action to check how the application handles multiple orders during a single user session.

### 1 Select a step to parameterize.

Expand **Copy of FlightOrder > Welcome: Mercury Tours > Find a Flight: Mercury**.

In the **toPort** row in the Keyword View, click the **Value** cell and then click the parameterization icon . The Value Configuration Options dialog box opens.



### 2 Enter the parameter information.

Select **Parameter** and confirm that the **DataTable** option is selected.


In the **Name** box, type ArrivalCity.

You are creating this data set exclusively for the **Copy of FlightOrder** action. Therefore, select **Current action sheet (local)**. Click **OK** to close the Value Configuration Options dialog box.


QuickTest automatically adds the **ArrivalCity** column to the **Copy of FlightOrder** data sheet and inserts **San Francisco** into the first cell of the **ArrivalCity** column.

### 3 Parameterize additional steps.

It is not logical for a single customer to fly to two different locations on the same dates, so it makes sense to parameterize the departure and arrival dates.

- ▶ In the **fromMonth** row in the Keyword View, click the **Value** cell and then click the parameterization icon . The Value Configuration Options dialog box opens. Select **Parameter** and confirm that the **DataTable** option is selected.

In the **Name** box, type `fromMonth`. Select **Current action sheet (local)** in the Value Configuration Options dialog box and click **OK**.

- ▶ In the **toMonth** row in the Keyword View, click the **Value** cell and then click the parameterization icon . The Value Configuration Options dialog box opens. Select **Parameter** and confirm that the **DataTable** option is selected.

In the **Name** box, type `toMonth`. Select **Current action sheet (local)** in the Value Configuration Options dialog box and click **OK**.

### 4 Enter the parameterization data in the Data Table.

In the Data Table, click the **Copy of FlightOrder** tab.

Type `Portland` in the second row of the **ArrivalCity** column.

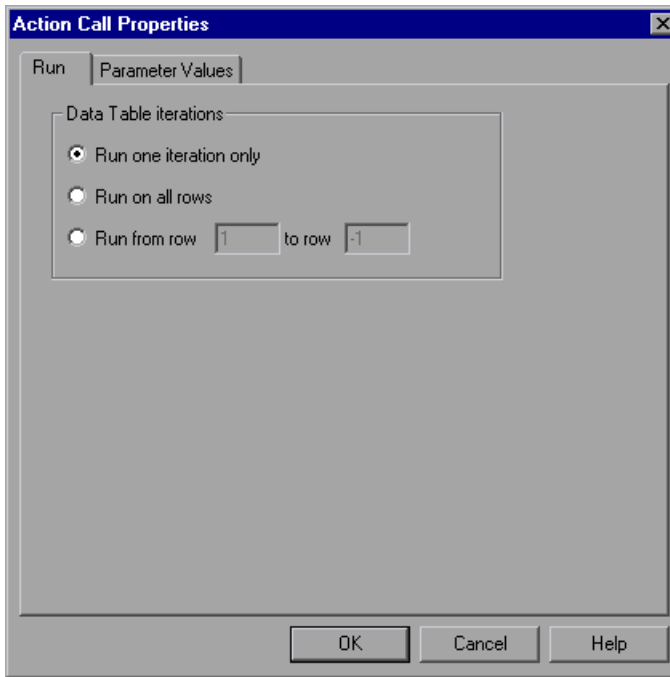
Type `October` in the second row of the **fromMonth** and **toMonth** columns.

Click an empty cell of the first or second row to save your changes.

### 5 Set the action to run all the data in the local Data Table.

By default, QuickTest only runs one iteration of each action in a test. You want QuickTest to use all the data in the **Copy of FlightOrder** tab.

Right-click the **Copy of FlightOrder** action in the Keyword View and choose **Action Call Properties**. The Action Call Properties dialog box opens.



Select **Run on all rows** in the Run tab. If a message box opens, read it and then click **OK** to close it. Click **OK** to close the Action Call Properties dialog box.

## 6 Save the test.



Choose **File > Save** or click the **Save** button.

You have now created a test that signs in to the Mercury Tours site, submits two flight orders: a flight from London to San Francisco from December 29 to December 31, and a flight from London to Portland from October 29 to October 31, and then returns to the starting point of the test by clicking the **Home** link.



## Running and Analyzing a Multi-Action Test

You will now run the **ActionB** test. The entire test will run only once, but the **FlightOrder** action will run twice; one time for each set of data in the **Copy of FlightOrder** data sheet.

### 1 Run the ActionB test.

Confirm that all Web browsers are closed.



Click **Run** or choose **Automation > Run**. The Run dialog box opens.

Select **New run results folder** and accept the default results folder name.

Click **OK**. When the test run is completed, the Test Results window opens.

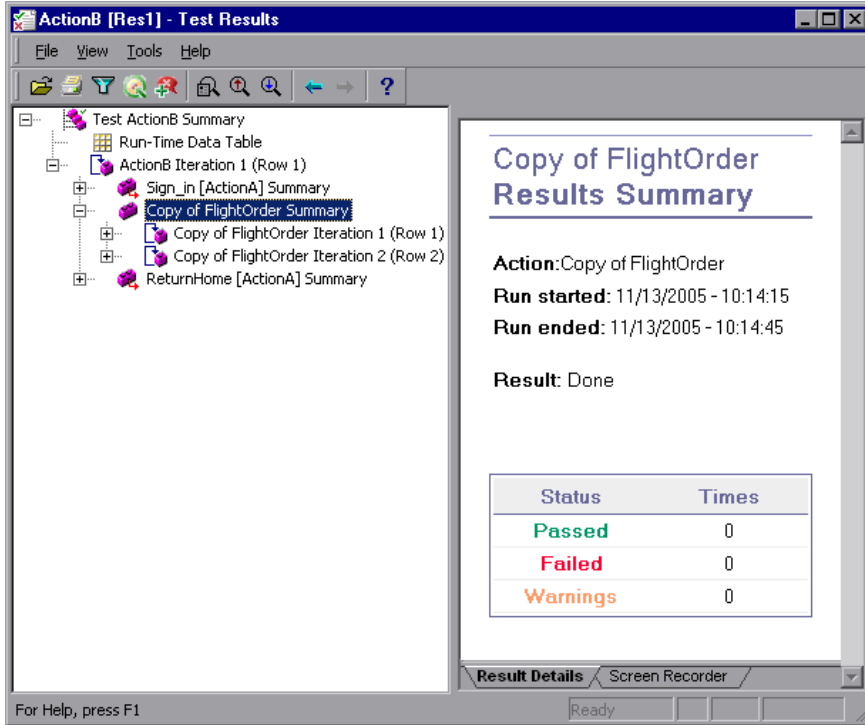
### 2 Examine the test results.

Examine the Results Summary. The test is marked as **Done**. This indicates that the test ran without any failures (there were no checkpoints to "pass").

Expand **Test ActionB Summary > ActionB Iteration 1 (Row 1)** in the Test Results window. Select the **Copy of FlightOrder Summary** action in the results tree.

## Lesson 8 • Dividing Tests into Multiple Actions

Note that within the summary, the results of both iterations are displayed.



### 3 Close the Test Results window.

Choose **File > Exit** to close the Test Results window.

# 9

---

## Where Do You Go from Here?

Now that you have completed the exercises in this tutorial, you are ready to apply the QuickTest concepts and the skills you learned to testing your own application.

In this lesson, you will learn about:	On page:
Testing Your Own Application—Getting Started	105

### Testing Your Own Application—Getting Started

In this tutorial, you learned about the basic tools needed for testing applications and Web sites.

We suggest that you follow the procedure outlined below when testing your own application.

#### **1 Plan your test.**

Decide how to organize your test. Consider what users will want to accomplish when deciding which operations to record. Confirm that your application and QuickTest are set to match the needs of your test.

Decide how to store the objects in your test. You can store the objects for each action in its corresponding local object repository, or you can store the objects for each action in one or more common (shared) object repositories. You can also use the same shared object repository for multiple actions.

If you are new to testing, you may want to use a local object repository for each action. This is the default setting, and all objects are automatically added to the local repository of each action.

If you are familiar with testing, it is probably most efficient to work in the shared object repository mode. In this mode, you can use shared object repositories for one or more actions. This means that object information is kept in one central location. When the objects in your application change, you can update them in one location for multiple actions in multiple tests.

For more information on object repositories, refer to the *QuickTest Professional User's Guide*.

## **2 Record your test.**

While recording (described in “Recording Tests” on page 23), follow the steps you expect users to perform as they navigate within your application or Web site.

## **3 Enhance your recorded test.**

- ▶ Add checkpoints (described in “Creating Checkpoints” on page 39) to search for specific values of a page, object, text string, or table.
- ▶ Replace fixed values in your test with parameters (described in “Parameterizing Tests” on page 59) to check how your application performs the same operations with multiple sets of data.
- ▶ Create output values (described in “Creating Output Values” on page 69) to retrieve data during a test run.
- ▶ Where applicable, use regular expressions (described in “Using Regular Expressions” on page 77) to increase the flexibility and adaptability of your tests.

You can further enhance your test with programming and conditional and loop statements, which add logic to your test. These are described in the *QuickTest Professional User's Guide*.

## **4 Debug your test.**

Debug your test to check that it operates smoothly and without interruption. For additional information on debugging, refer to the *QuickTest Professional User's Guide*.

**5 Run your test.**

Run your test on your application or Web site (described in “Running and Analyzing Tests” on page 33) to check that the application functions as expected.

**6 Analyze the test results.**

Examine the results of your test to pinpoint defects in your application. (Refer to the appropriate sections of this tutorial to understand what to look for in the test results.)

**7 Report defects.**

If you have Quality Center installed, you can submit any defects discovered to a Quality Center database. (Quality Center is the Mercury solution for test management. For additional information, refer to the *QuickTest Professional User's Guide* and the documentation included with Quality Center.)

**Lesson 9 • Where Do You Go from Here?**

---

# Index

## A

Action Properties dialog box 102  
Action toolbar 19  
actions 83–104

- analyzing test results 103
- calling 93
- copying 96
- creating 84
- inserting existing 93
- parameterizing 99
- running tests 103
- splitting 87

Active Screen 20  
analyzing test results 35–38

- actions 103
- checkpoints 53
- output values 75
- parameterized tests 65
- regular expressions 81

automated testing, benefits 13  
Automation toolbar 18

## C

calling actions 93  
Checkpoint Properties dialog box 44, 51  
checkpoints

- analyzing test results 53
- creating 39–58
- page 46
- running tests with 53
- standard 42, 57
- table 50
- text 48

conventions, typographical 12  
copying actions 96  
creating actions 84

creating checkpoints 39–58  
creating tests

- procedure 105–107
- with multiple actions 83–104
- with output values 69

## D

Data Table 20

- assigning parameter values 62

Debug toolbar 19  
Debug Viewer pane 20  
documentation updates 11  
documentation, online 8

## E

Edit toolbar 18

## I

Information pane 20  
input parameters 59–67  
Insert toolbar 18

## K

Keyword View 30  
Knowledge Base 10

## M

manual testing, drawbacks 13  
menu bar, QuickTest window 17  
Mercury Best Practices 10  
Mercury Customer Support Web site 10  
Mercury Home Page 10  
Mercury Tours sample application 21

## Index

Missing Resources pane 20  
modifying parameterized tests 63  
multiple actions 83–104

## O

Object Selection - Checkpoint Properties  
    dialog box 43, 46, 50  
online documentation 8  
online resources 10  
output values 69–76

## P

Page Checkpoint Properties dialog box 47  
page checkpoints  
    creating 46  
    results 54  
parameter values, in Data Table 62  
parameterized tests  
    modifying 63  
    running and analyzing results 65  
parameterizing  
    actions 99  
    tests 59–67  
parameters  
    definition 60  
    input 59–67

## Q

QuickTest window 16  
    Action toolbar 19  
    Active Screen 20  
    Automation toolbar 18  
    Data Table 20  
    Debug toolbar 19  
    Debug Viewer pane 20  
    Edit toolbar 18  
    Information pane 20  
    Insert toolbar 18  
    menu bar 17  
    Missing Resources pane 20  
    Standard toolbar 17  
    status bar 20  
    Test pane 20  
    title bar 17, 20

Tools toolbar 17  
View toolbar 17

## R

Readme 8  
Record and Run Settings dialog box 25  
recording tests 23  
regular expressions 77–82  
Run dialog box 34  
running tests 33–38  
    parameterized tests 65  
    with actions 103  
    with checkpoints 53  
    with output values 75  
    with regular expressions 81

## S

sample application, Mercury Tours 21  
Split Action dialog box 87  
standard checkpoints  
    creating 42  
    results 57  
Standard toolbar 17  
status bar, QuickTest window 20

## T

table checkpoints  
    creating 50  
    results 56  
Test pane 20  
test results  
    actions 103  
    analyzing 35–38  
    checkpoints 53  
    output values 75  
    regular expressions 81  
Test Results window 35  
testing applications, how to begin 105–107  
testing process 14  
tests  
    analyzing results 35–38  
    procedure for creating 105–107  
    recording 23  
    running 33–38



- Text Checkpoint Properties dialog box 49
- text checkpoints
  - creating 48
  - results 58
- Text Output Value Properties dialog box 71
- title bar, QuickTest window 17, 20
- toolbars
  - Action 19
  - Automation 18
  - Debug 19
  - Edit 18
  - Insert 18
  - Standard 17
  - Tools 17
  - View 17
- Tools toolbar 17
- typographical conventions 12

**U**

- updates, documentation 11

**V**

- values, output 69–76
- View toolbar 17

