

HP QuickTest Professional

Software Version: 11.00

Tutorial

Document Release Date: October 2010
Software Release Date: October 2010



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© 1992 - 2010 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Intel®, Pentium®, and Intel® Xeon™ are trademarks of Intel Corporation in the U.S. and other countries.

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, Windows NT®, and Windows® XP are U.S registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

Unix® is a registered trademark of The Open Group.

SlickEdit® is a registered trademark of SlickEdit Inc.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support web site at:

<http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Table of Contents

Welcome to The QuickTest Professional Tutorial.....	7
QuickTest Professional Tutorial Overview	7
Documentation Library Contents.....	9
Additional Online Resources.....	12
Lesson 1: Introducing QuickTest.....	13
Benefits of Automated Testing	14
Testing Process.....	15
QuickTest Window	17
Lesson 2: Analyzing Your Application.....	23
Getting to Know Your Application	24
Exploring the Mercury Tours Web Site	25
Creating an Action	27
Lesson 3: Creating Object Repositories	33
Introducing Test Objects and Object Repositories.....	33
Learning Objects in Your Application	35
Using Multiple Object Repositories.....	44
Lesson 4: Creating Functions and Function Libraries	49
Introducing Functions and Function Libraries	49
Creating a Function.....	50
Associating the Function Library with Your Test	52
Lesson 5: Creating a Test	55
Preparing to Create a Test	55
Adding Steps to the Login Action	56
Analyzing the Login Action in the Keyword View	62
Adding the Remaining Steps to Your Test	64
Dividing an Action into Two Actions	78
Lesson 6: Running and Analyzing Tests.....	81
Running a Test.....	81
Analyzing Run Results	83

Table of Contents

Lesson 7: Creating Checkpoints and Using Functions.....	87
Understanding Checkpoint Types	88
Checking Objects	90
Checking Pages.....	93
Checking Tables	95
Checking Text	100
Managing Checkpoints in the Object Repository.....	104
Running and Analyzing a Test with Checkpoints	105
Performing a Check Using a Function	111
Lesson 8: Parameterizing Tests.....	117
Defining a Data Table Parameter	117
Adding Parameter Values to a Data Table.....	120
Modifying Steps Affected by Parameterization	121
Parameterizing an Action	123
Running and Analyzing a Parameterized Test	127
Lesson 9: Where Do You Go from Here?	129
Modifying QuickTest Settings to Save Disk Space	129
Testing Your Own Application—Getting Started	130

Welcome to The QuickTest Professional Tutorial

This chapter includes:

- "QuickTest Professional Tutorial Overview" on page 7
- "Documentation Library Contents" on page 9
- "Additional Online Resources" on page 12

QuickTest Professional Tutorial Overview

Welcome to the QuickTest Professional 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 run results. After completing the tutorial, you can apply the skills you have learned to your own application or Web site.

Using QuickTest Professional with Business Process Testing

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, see the *HP QuickTest Professional for Business Process Testing User Guide* and the *HP Business Process Testing User Guide*.

Note: Unless otherwise specified, references to **Quality Center** in this guide apply to all currently supported versions of Quality Center and HP ALM. Note that some features and options may not be supported in the specific edition of Quality Center or HP ALM that you are using.

For a list of the supported versions of Quality Center or HP ALM, see the *HP QuickTest Professional Product Availability Matrix*, available from the Documentation Library Home page or the root folder of the QuickTest Professional DVD.

For details on Quality Center or HP ALM editions, see the *HP Quality Center User Guide* or the *HP Application Lifecycle Management User Guide*.

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.

Documentation Library Contents

This tutorial is part of the QuickTest Professional Documentation Library. The Documentation Library provides a single-point of access for all QuickTest Professional documentation.

You can access the Documentation Library by using the following:

- Select **Help > QuickTest Professional Help**.
- In the Start menu, select **Program Files > HP QuickTest Professional > Documentation > HP QuickTest Professional Help**.
- Click in selected QuickTest windows and dialog boxes or press F1.
- View a description, syntax, and examples for a QuickTest test object, method, or property by placing the cursor on it and pressing F1.

The Documentation Library includes the following:

Type	Included Documentation
Getting Started Documentation	<ul style="list-style-type: none">➤ Readme provides the latest news and information about QuickTest. Select Start > Programs > HP QuickTest Professional > Readme.➤ HP QuickTest Professional Installation Guide explains how to install and set up QuickTest. Select Help > QuickTest Professional Help and click the link to the Installation Guide from the Documentation Library Home page.➤ HP QuickTest Professional Tutorial teaches you basic QuickTest skills and shows you how to design tests for your applications. Select Help > QuickTest Professional Tutorial.➤ Product Feature Movies provide an overview and step-by-step instructions describing how to use selected QuickTest features. Select Help > Product Feature Movies.➤ What's New provides an overview of the features, enhancements and supported environments that are new in the current version of <i>QuickTest</i>. Choose Help > What's New.

Type	Included Documentation
Feature Documentation	<p>QuickTest Professional Help includes:</p> <ul style="list-style-type: none">➤ Home provides links to the Documentation Library guides in each available format (Help, PDF, and/or HTML).➤ What's New in QuickTest Professional describes the newest features, enhancements, and supported environments in the latest version of QuickTest.➤ HP QuickTest Professional User Guide describes how to use QuickTest to test your application.➤ HP QuickTest Professional for Business Process Testing User Guide provides step-by-step instructions for using QuickTest to create and manage assets for use with Business Process Testing.➤ HP QuickTest Professional Add-ins Guide describes how to work with supported environments using QuickTest add-ins, and provides environment-specific information for each add-in.➤ HP QuickTest Professional Object Model Reference describes QuickTest test objects, lists the methods and properties associated with each object, and provides syntax information and examples for each method and property.

Type	Included Documentation
Reference Documentation	<p>► HP QuickTest Professional Advanced References contains documentation for the following QuickTest COM and XML references:</p> <ul style="list-style-type: none"> ► HP QuickTest Professional Automation Object Model 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. ► HP QuickTest Professional Run Results Schema documents the run results XML schema, which provides the information you need to customize your run results. ► HP QuickTest Professional Test Object Schema documents the test object XML schema, which provides the information you need to extend test object support in different environments. ► HP QuickTest Professional Object Repository Schema documents the object repository XML schema, which provides the information you need to edit an object repository file that was exported to XML. ► HP QuickTest Professional Object Repository Automation documents the Object Repository automation object model, which provides the information you need 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.

Additional Online Resources

Sample applications. The following sample applications are the basis for many examples in this guide:

- **Mercury Tours sample Web site.** The URL for this Web site is newtours.demoaut.com.
- **Mercury Flight application.** To access from the Start menu, select **Program Files > HP QuickTest Professional > Sample Applications > Flight.**

Troubleshooting & Knowledge Base accesses the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help > Troubleshooting & Knowledge Base**. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpsoftwaresupport.

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport user ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

HP Software Web site accesses 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 > HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

1

Introducing QuickTest

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

It is recommended to work with an online copy of this tutorial because there are sections in which you will be asked to copy and paste test steps into QuickTest. You can open a PDF of this tutorial from your <QuickTest Professional installation folder>\help folder or by choosing **Help > Printer-Friendly Documentation** in QuickTest. You can access the version of this tutorial included with the help, by choosing **Help > QuickTest Professional Tutorial**.

This lesson includes:

- Benefits of Automated Testing on page 14
- Testing Process on page 15
- QuickTest Window on page 17

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.

Benefits of Automated Testing	
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 the following main phases:

1 Analyzing your application

The first step in planning your test is to analyze your application to determine your testing needs.

- What are your application's development environments (for example Web, Java, or .NET)? You will need load QuickTest add-ins for these environments to enable QuickTest to identify and work with the objects in your application.
- What business processes and functionality do you want to test? To answer this, think about the various activities that customers perform in your application to accomplish specific tasks.
- Consider how to divide these business processes into smaller units. You will be creating actions based on these tasks. Smaller and more modular actions make your tests easier to read and follow, and help ease maintenance in the long run.

At this stage, you can already begin creating test skeletons and adding actions to them.

2 Preparing the testing infrastructure

Based on your testing needs, you determine what resources are required and create these resources, accordingly. Resources include shared object repositories containing test objects (which are representations of the objects in your application), function libraries containing functions that enhance QuickTest functionality, and so on.

You also need to configure QuickTest settings so that QuickTest will perform the tasks you need, such as displaying a results report every time you run a test.

3 Building your tests and adding steps to them

After the testing infrastructure is ready, you can begin building your tests. You can create one or more empty tests and add actions to them to create the testing skeletons. You associate your object repositories with the relevant actions, and associate your function libraries with the relevant tests, so that you can insert steps using keywords. You may also need to configure test preferences at this point.

4 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 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.

5 Debugging, running, and analyzing your test

You debug a test to ensure that it operates smoothly and without interruption. After the test is working correctly, you run it to check the behavior of your application. While running, QuickTest opens the application and performs each step in your test.

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

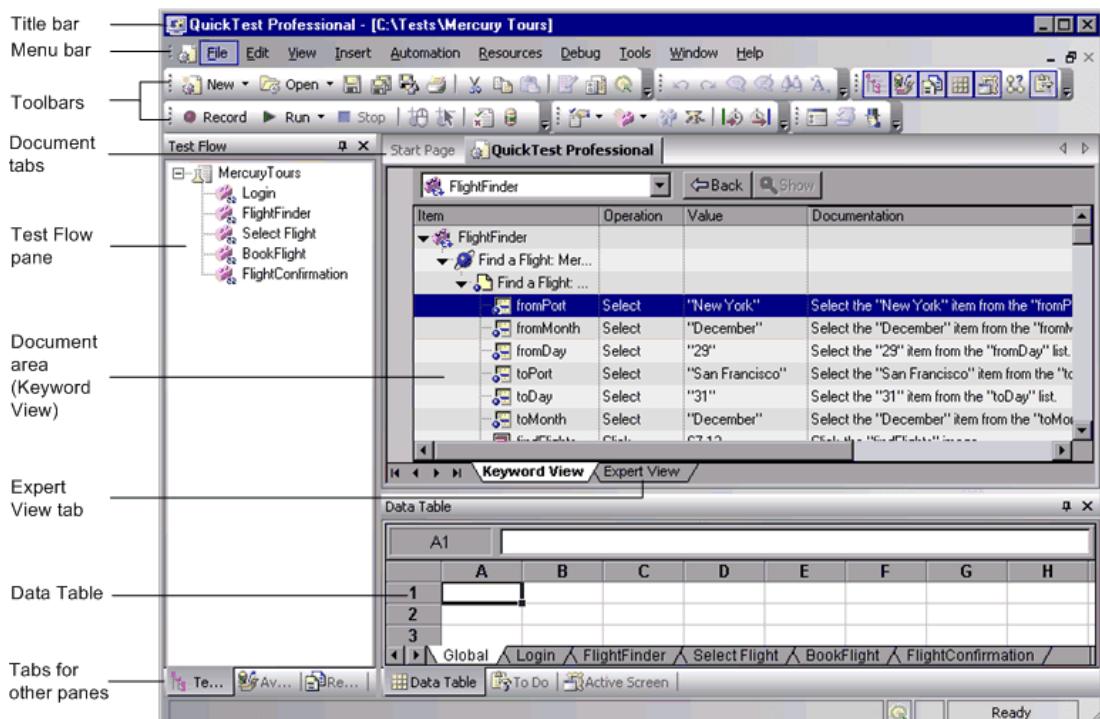
6 Reporting defects

If you have Quality Center installed, you can report the defects you discover to a database. Quality Center is the HP 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 create a test, with all toolbars and the Data Table and Resources panes displayed:



Document Area

The document area of the QuickTest window can display the following:

- **Test.** Enables you to create, view, and modify your test in the Keyword View or Expert View:
- **Keyword View.** Contains each step, and displays the object hierarchy, in a modular, icon-based table.
- **Expert View.** Contains each step as a VBScript line. In object-based steps, the VBScript line defines the object hierarchy.
- **Function Library.** Enables you to create, view, and modify functions (operations) for use with your test. Function libraries open as separate tabs or windows in the documentation area, similar to tests.
- **Start Page.** Welcomes you to QuickTest and provides links to Process Guidance. You can use the shortcut buttons to open new and existing documents.
- **Document tabs.** Enables you to navigate open documents in the document area by selecting the tab of the document you want to activate (bring into focus).

Toolbars and Menus

In addition to the document area, the QuickTest window contains the following key elements:

- **QuickTest title bar.** Displays the name of the active document. If changes have been made since it was last saved, an asterisk (*) is displayed next to the document name in the title bar.
- **Menu bar.** Displays menus of QuickTest commands.
- **Standard toolbar.** Contains buttons to assist you in managing your testing documents.



- **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 running and analyzing your test.



- **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: For information on all of the toolbar and menu options, see the *HP QuickTest Professional User 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 more information, see the *HP QuickTest Professional User Guide*.

- **Status bar.** Displays the status of the QuickTest application and other relevant information.

Panes (Available Also From the View Menu)



- **Active Screen.** Provides a snapshot of your application as it appeared when you performed a certain step during a recording session. (You can display the Active Screen by choosing **View > Active Screen**, or by clicking the **Active Screen** button.)



- **Available Keywords.** Displays all the keywords available to your test. Enables you to drag and drop objects or calls to functions into your test. (You can display the Available Keywords pane by choosing **View > Available Keywords**, or by clicking the **Available Keywords** button.)



- **Data Table.** Assists you in parameterizing your test. (You can display the Data Table pane by choosing **View > Data Table**, or by clicking the **Data Table** button.)

- **Debug Viewer.** 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**.)

- **Information.** Displays a list of syntax errors found in your test and function library scripts. (You can display the Information pane by choosing **View > Information**.)



- **Missing Resources.** Provides a list of the resources that are specified in your test but cannot be found, such as missing Recovery Scenarios and missing Function Libraries. (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.)



- **Process Guidance.** Displays two panes that provide procedures and descriptions on how to best perform specific processes, such as creating a test in QuickTest. The Process Guidance Activities pane lists the activities that you can perform, such as adding steps to a test. The Process Guidance Description pane describes the tasks that you need to perform for a selected activity. Your organization may also provide you with process guidance that is accessible from these panes. (You can display the Process Guidance panes by choosing **View > Process Guidance**, or by clicking the **Process Guidance panes** button.)



- **Test Flow.** Displays the hierarchy of actions and action calls in the current test, and shows the order in which they are run. (You can display the Test Flow pane by choosing **View > Test Flow**, or by clicking the **Test Flow Pane** button.)



- **Resources.** Displays all the resources associated with your current test and enables you to manage these resources. (You can display the Resources pane by choosing **View > Resources**, or by clicking the **Resources Pane** button.)



- **To Do.** Displays and enables you to manage the tasks defined for the current test. The To Do pane also displays the TODO comment steps of the test's actions or currently open function libraries. (You can display the To Do pane by choosing **View > To Do**, or by clicking the **To Do Pane** button.)

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

You can also execute many menu commands using shortcut keys. For more information, see the *QuickTest at a Glance* chapter in the *HP QuickTest Professional User Guide*.

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

2

Analyzing Your Application

In this lesson, you will analyze an application to determine what needs to be tested, and you will learn to create an action in QuickTest. An action is a logical unit in your test that focuses on testing a particular part of your application.

This lesson includes:

- Getting to Know Your Application on page 24
- Exploring the Mercury Tours Web Site on page 25
- Creating an Action on page 27

Getting to Know Your Application

Before you begin creating tests, you need to determine what exactly you want to test in your application. To do this, you need to analyze your application in terms of its business processes—the distinct activities that customers perform in your application to complete particular tasks.

For the purpose of this tutorial, you need to become familiar with the Mercury Tours Web site. Mercury Tours is a sample Web application that simulates a Web-based flight information and reservation service. You can access the Mercury Tours Web site using any Web browser, but you can perform the tutorial using only Microsoft Internet Explorer.

As you navigate through this application, consider the following:

- How is the application divided? Are there separate areas for each activity?
What are these areas? What are these activities? What are the expected outcomes for each activity?
- What screens need to be tested in each of these areas? What objects need to be tested in each screen?
- What activities might be used in more than one scenario? (For example, logging on to the application.) Think of these as "reusable" parts.

Analyzing your application in terms of its separate parts can help you to design your tests so that they comprise small, modular units, which are easier to understand, debug, and maintain over time.

- What business processes might users perform using this application? One scenario discussed in this tutorial is reserving flights. Other scenarios might include making cruise or hotel reservations, or even registering on the Web site.
- Consider which development environments need to be supported for testing purposes. QuickTest provides add-ins to support many testing environments. You load these add-ins when you open QuickTest. When working with this tutorial, you need to load the Web add-in because Mercury Tours is a sample Web application. Your application might require you to load Java or other add-ins.

Exploring the Mercury Tours Web Site

Before you begin this tutorial, open the Mercury Tours Web site and explore it for a while. As you navigate the site and familiarize yourself with its content, consider the questions posed in the previous section.

To run Mercury Tours:

1 Launch the Mercury Tours Web site.

In Microsoft Internet Explorer, type the following URL:

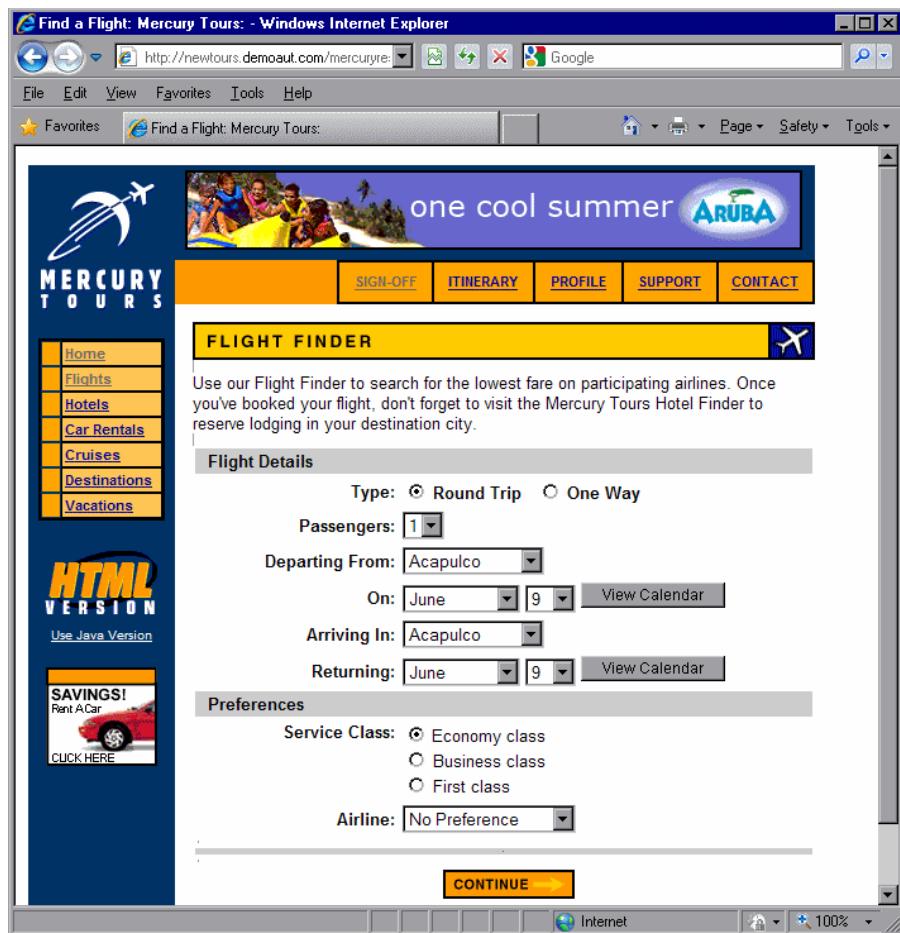
<http://newtours.demoaut.com>

The Mercury Tours home page opens.

2 Log in to Mercury Tours.

In the Find a Flight area, enter **tutorial** in both the **User Name** and **Password** boxes and click **Sign-In**.

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.

As you navigate the application, think about what you might want to test, and which pages and objects need to be included in each test that you might create.

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 in red) in the **Passengers** and **Credit Card** sections. (You may enter fictitious information.)

4 End your Mercury Tours session.

When you finish exploring the Mercury Tours Web site, 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 **tutorial** in the **User Name** and **Password** boxes 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 Web site using QuickTest!

Creating an Action

Each QuickTest test comprises calls to actions. Actions are 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.

QuickTest lets you insert several types of actions into tests. An **internal** action is an action that is stored in the local test (also known as a source test). An **external** action is a referenced call to an action that is stored in a different test. For example, you may have an action that you want to use in multiple tests—you would store the action as an internal action in one test and insert calls to that action from other tests. In the other tests the action would be available as an external action.

When you insert a call to a new action, it is **reusable** by default, enabling you to insert calls to the action from any test. Reusable actions are especially useful when you have a process that you may need to include in several tests, such as logging in to your application, or that you may need to insert several times in the same test, such as filling in user credential dialog boxes that open whenever a user tries to access a secure part of your application. Reusable actions can be internal or external actions.

When you insert calls to existing (reusable) actions, they are read-only in the calling test. You modify them in the test in which they are stored.

Inserting calls to existing actions enables you to create and enhance your action's test steps only once and then to call the same action from several tests. This makes it easier to maintain tests, because when your application changes you only have to update the existing action stored with the original test. This also helps ease test maintenance because when you modify these actions, all tests containing calls to these actions are updated. This enables you to create and enhance your action's test steps only once and then insert calls to the existing actions as needed from any test. If you plan to use the same or similar actions in many tests, you might consider creating a repository test to store your reusable actions. Then you can insert calls to these existing actions from other tests, linking these actions to the source test.

You can also insert a copy of an action in your tests if you plan to make changes to the action steps. This inserts the action as an **internal** action in the current test. Copies of actions are not linked to the source test, so that 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.

If you want to prevent an action from being reused in any test, you can make it **non-reusable**. Non-reusable actions are internal actions that can be used only once.

In this section, you will create a new test and rename an action so that anyone viewing this test will know the purpose of the action. You will use this action in subsequent lessons.

1 Start QuickTest.

- If QuickTest is not currently open, choose **Start > Programs > HP QuickTest Professional > HP 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. While QuickTest loads your selected add-ins, the QuickTest splash screen is displayed. This may take a few seconds.

- 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 Create a new test in one of the following ways:



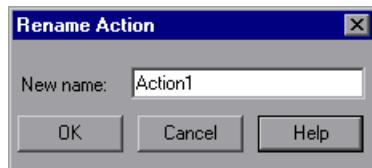
- In the Start Page window, in the **Welcome!** area, click the **New Test** button.
- In the QuickTest main window, choose **File > New > Test**, or click the **New** button down arrow and choose **Test**. If a test is already open, clicking the **New** toolbar button opens a new test.

A blank test opens.

3 Rename Action1 so that it has a logical name.



- a** Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b** In the Test Flow pane, select **Action1** and choose **Edit > Action > Rename Action**. In the confirmation box, click **Yes**.

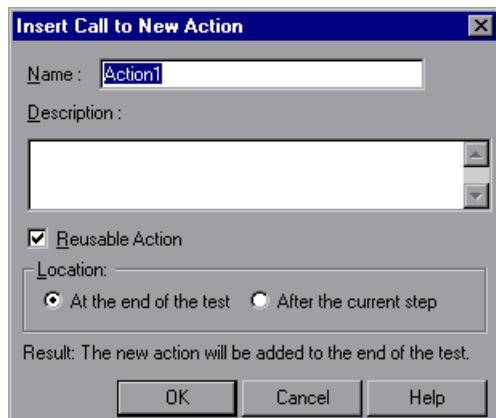


- c** In the **New name** box, enter **Login**.
- d** Click **OK** to save the change.

4 Create a new action named FlightFinder.



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



- b** Enter **FlightFinder** in the **Name** box, confirm that both **Reusable Action** and **At the end of the test** are selected, and click **OK**. The dialog box closes and the **FlightFinder** action is added to the end of your test.

5 Create calls to additional actions.

Repeat step 4 to add the following actions:

- **SelectFlight**
- **BookFlight**

Your test now contains all of the actions needed to begin this tutorial.

6 Save your test.



- a Choose **File > Save** or click the **Save** button. The Save dialog box opens to the Tests folder.
- b Create a folder named Tutorial, select it, and click **Save**.
- c Enter MercuryTours in the **File name** field.
- d Confirm that **Save Active Screen files** is selected.
- e Click **Save**. The test name (MercuryTours) 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.

Different Ways to Insert Calls to Actions

You can insert calls to actions in your test at any time (except during a test run). In the previous section, you inserted a call to a new action using the first option listed below. You can use any 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.

This option inserts a new, empty action in your test.

(You can also click the down arrow and choose **Call to Copy of Action** or **Call to Existing Action**, described below.)



- **Edit > Action > Split Action** or use the **Split Action** button.

This option divides an existing action into two separate actions at the step you specify.

- **Insert > Call to Copy of Action** or right-click an action and choose **Insert Call to Copy of Action**.

This option inserts a copy of an existing action in your test. The action is not linked to the source test from which you copied it.

- **Insert > Call to Existing Action** or right-click an action and choose **Insert Call to Existing Action**.

This option inserts a call to an existing action in your test. You can insert a call to an action that is stored in the current test (if you want to insert the same action twice, for example), or you can insert a call to an action in another test. The action is linked to the source test from which you copied it.

3

Creating Object Repositories

This lesson introduces test objects and run-time objects, and describes how to create shared object repositories.

This lesson includes:

- Introducing Test Objects and Object Repositories on page 33
- Learning Objects in Your Application on page 35
- Using Multiple Object Repositories on page 44

Introducing Test Objects and Object Repositories

Before you create a test, you need to set up the resources that will be used by that test. One of the most important resources for any test is the object repository. An object repository is a storehouse for the test objects (and other object types) used in your test.

Test objects are stored representations of the actual objects (or controls) in your application. QuickTest creates test objects by learning a select set of the properties and values of the objects in your application. QuickTest uses the information it learns to uniquely identify the run-time objects in your application.

Each test object is part of a test object hierarchy. For example, a Link object may be part of a Browser/Page/Link hierarchy. Top-level objects, such as Browser objects, are known as container objects, as they can contain lower-level objects, such as Frame or Page objects.

Note: Object repositories can also include checkpoint objects. These object types are covered in Lesson 7, "Creating Checkpoints and Using Functions."

Run-time objects are created and maintained during a QuickTest run session by the object creator (for example, Microsoft for Microsoft Internet Explorer objects, Mozilla for Mozilla Firefox objects, the product developer for ActiveX objects, and so on). During a run session, QuickTest performs the specified test object method on the run-time object. Run-time objects are not stored in an object repository, as they are available only during the run session.

Later in this lesson, you will use the Object Spy dialog box to view the properties and operations of objects in your application.

Object Repositories

QuickTest can store the test objects it learns in two types of object repository files: shared and local.

A **shared object repository** contains test objects that can be used in multiple actions. This versatility makes it the preferred repository type for storing and maintaining test objects. By associating a shared object repository with an action, you make the test objects in that repository available for use in the action. Any maintenance you perform on an object in a shared object repository is reflected in any step using that object.

A **local object repository** stores test objects that can be used in one specific action. The objects in this type of repository cannot be used in any other action. Local object repositories are useful for backing up your test objects, or for learning one or several objects.

When you create an object repository, try to include only the objects you need for testing purposes. This keeps the object repository relatively small and helps makes maintenance and object selection easier. Also, make sure that you provide logical names so that others can easily choose the correct objects when creating or modifying tests.

During a run session, QuickTest references the test objects stored in the associated object repositories in order to perform operations on the corresponding objects in your application.

Learning Objects in Your Application

In this section, you will use the QuickTest Navigate and Learn option to add objects from the Mercury Tours Web site to a shared object repository. Then you will associate the object repository with your test.

1 Start QuickTest.

If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.

2 If it is not currently open, open the MercuryTours test you created in Lesson 2, "Creating an Action", in one of the following ways:

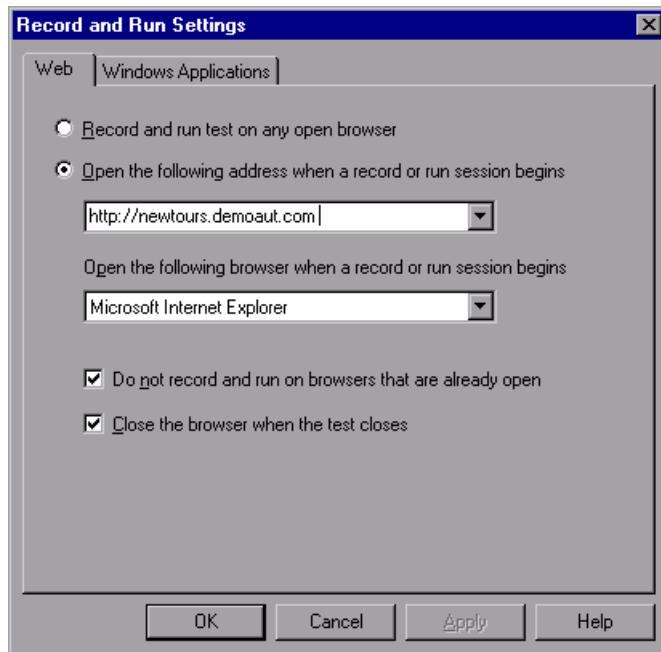
- Choose **File > Open > Test**, navigate to the **MercuryTours** test and click the **Open** button.
- Click the **Open** button or the **Open** button down arrow. The Open Test dialog box opens. Navigate to the **MercuryTours** test and click the **Open** button.



3 Set the learn settings for QuickTest.

- a Choose **Automation > Record and Run Settings**. The Record and Run Settings dialog box opens.
- b In the Web tab, select **Open the following address when a record or run session begins**.
- c Confirm that the URL in the first box is <http://newtours.demoaut.com>. In the second box, choose a browser on which to run your test session. Microsoft Internet Explorer is the browser used in this tutorial.

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



- e** Click **OK**. When you run the test, your browser will open to the Mercury Tours Web site. When the run session ends, your browser will close.

4 Open the Mercury Tours Web site.

Open Internet Explorer and browse to the Mercury Tours Web site: <http://newtours.demoaut.com>

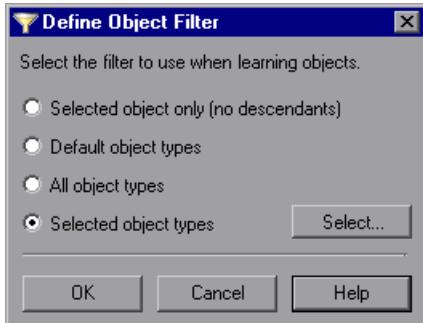
5 View the properties and operations for some of the objects in the Mercury Tours Web site.

- a** In QuickTest, choose **Tools > Object Spy**. The Object Spy dialog box opens.
- b** Drag the Object Spy dialog box to the right side of your screen. This will enable you to see the objects you want to spy on more clearly. Verify that the **Keep Object Spy on top while spying** button is pressed.

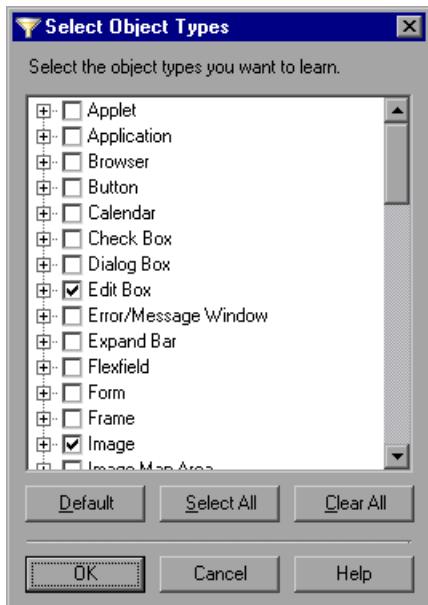




- c** Click the pointing hand button. QuickTest is hidden, and the Object Spy dialog box is displayed over the Mercury Tours Web site Welcome page.
 - d** Move the mouse cursor over the various objects on the page and watch what happens in the Object Spy dialog box.
 - e** Position the mouse cursor in the **User Name** edit box (but do not click it). The test object hierarchy of that object is displayed. In the hierarchy box, notice that the name of the object is **WebEdit**, which is its object class type.
 - f** Click the **User Name** edit box. QuickTest opens again. In the hierarchy box, notice that the object class and name are now displayed (**WebEdit:userName**).
 - g** Click **Close** to close the Object Spy dialog box.
- 6 Start the Navigate and Learn process by setting up the Define Object Filter.**
- a** In QuickTest, choose **Resources > Object Repository Manager**. The Object Repository Manager window opens.
 - b** In the Object Repository Manager window, choose **Object > Navigate and Learn**. Both QuickTest and the Object Repository Manager are hidden. The first time you use Navigate and Learn, a message box displaying a tooltip opens. Click **OK** to display the Navigate and Learn toolbar.
 - c** In the Navigate and Learn toolbar, click the **Define Object Filter** button. The Define Object Filter dialog box opens.



- d In the Define Object Filter dialog box, select the **Selected object types** radio button and click **Select**. The Select Object Types dialog box opens.



- e Click the **Clear All** button to clear all of the check boxes. Then select the **Edit Box** and **Image** check boxes and click **OK**. This instructs QuickTest to learn only the objects that are needed for your test and add them to the object repository. (This helps keep object repository maintenance to a minimum.)
- f In the Define Object Filter dialog box, click **OK**.

7 Learn the selected object types from the Mercury Tours Web site Welcome page.

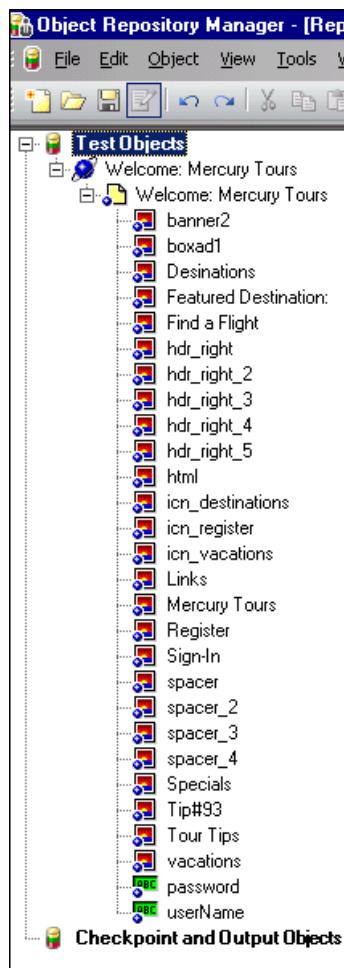
In this step, you instruct QuickTest to learn all of the objects in the Mercury Tours Web site Welcome page that match your filter and to add them to a shared object repository.

- a** In the Mercury Tours Web site Welcome page, click the title bar to bring it into focus.
- b** In the Navigate and Learn toolbar, click the **Learn** button. The Web page flickers and the Adding Objects message box is displayed as QuickTest begins adding representations of the objects on the Web page to a new object repository. Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs. When the flashing stops and the Adding Objects box closes, Navigate and Learn has finished adding the objects to the object repository.
- c** In the Navigate and Learn toolbar, click the **Close** button. The Navigate and Learn toolbar closes, and QuickTest and the Object Repository Manager window are visible again.

8 Remove unneeded objects from the object repository.

In the previous step, QuickTest added all of the objects that match your filter selection to the object repository. For this test, you need only a few objects, so in this step, you will delete the extraneous objects.

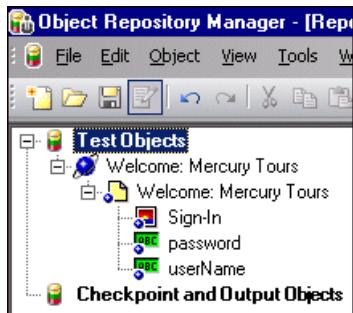
- a In the Object Repository Manager window, expand the objects in the **Test Objects** tree. You should see a list of objects similar to the following:





- b** Delete all of the test objects EXCEPT for **Sign-In** (image), **password** (edit box), and **userName** (edit box), by selecting the objects and choosing **Edit > Delete** or clicking the **Delete** button. You can select multiple objects using the SHIFT and CTRL keys. Click **Yes** in the confirmation box.

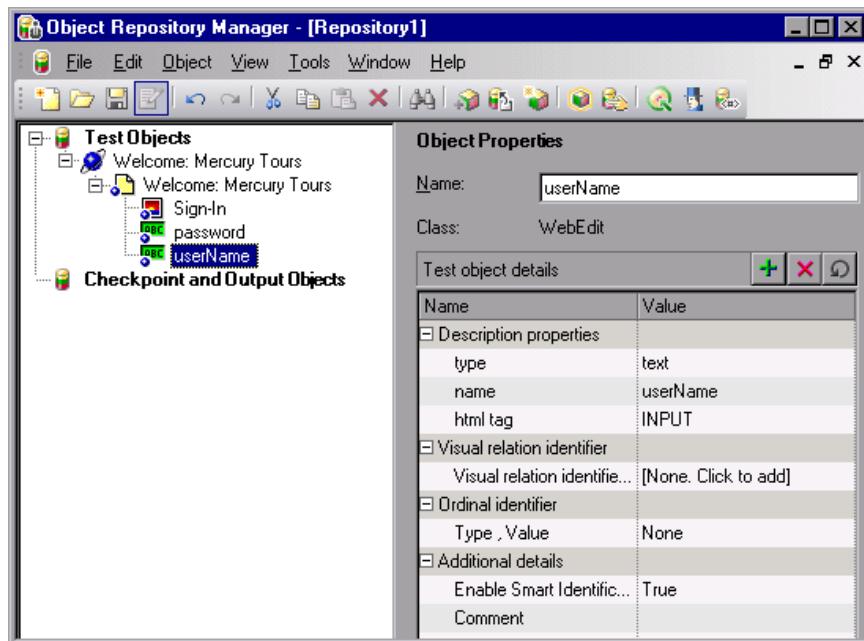
After you delete the extraneous objects, your tree should resemble the following:



Tip: If you delete **Sign-In**, **password**, or **userName** by mistake, click **Undo** to restore the deleted objects.

9 See what QuickTest learned about one of the objects.

In the **Test Object** tree, highlight the **userName** object. In the right pane, notice its object properties. These are the descriptive properties that QuickTest will use to identify the object during a run session.

**10 Save the object repository.**

- In the Object Repository Manager window, choose **File > Save** or click the **Save** button. The Save Shared Object Repository dialog box opens.
- Browse to the location in which you want to save your object repositories, create a folder named **Tutorial_ObjectRepositories** and double-click it. In the **File Name** box, enter **MercuryToursLogin** and click **Save**.

11 Associate the object repository with the Login action.

- a** Switch to the QuickTest window and open the Resources pane, if it is not already open, by choosing **View > Resources** or clicking the **Resources** button.
- b** In the Resources pane, right-click the Login action and choose **Associate Repository with Action**. The Open Object Repository dialog box opens.
- c** Locate and select the **MercuryToursLogin.tsr** shared object repository and click **Open**. The object repository is now associated with the **Login** action, and is displayed in the Resources pane as a child of that action.

Associating an object repository with an action enables you to use any object from that repository in any step in the associated action.

12 Save your test.

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

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

Congratulations! You have created a shared object repository and associated it with your test's action.

Using Multiple Object Repositories

In the previous section, you created a shared object repository for the Welcome page of the Mercury Tours Web site. Now, you will create a shared object repository for each of the remaining pages in the site.

Tip: It is recommended to always create a separate shared object repository for each page in your Web site or each area of your application. This makes it easier to find the correct object when adding or modifying test steps or when performing maintenance tasks when objects are added, removed, or modified in your application.

For the purposes of this tutorial, you will instruct QuickTest to learn all of the objects on each page. At this point, you do not need to associate the object repositories with a specific action.

1 If it is not still open, log in to the Mercury Tours Web site.

- a** Open Internet Explorer and browse to the Mercury Tours Web site:
<http://newtours.demoaut.com>
- b** In the **User Name** and **Password** boxes, type **tutorial**.
- c** Click **Sign-In**. The Flight Finder page opens.

2 Create a new shared object repository.

- a** If the Object Repository Manager is closed, open it by choosing **Resources > Object Repository Manager** in QuickTest. The Object Repository Manager window opens.
- b** In the Object Repository Manager window, choose **File > New**. A blank shared object repository opens.

3 Start the Navigate and Learn process by setting up the Define Object Filter.

- a** In the Object Repository Manager window, choose **Object > Navigate and Learn**. Both QuickTest and the Object Repository Manager are hidden.
- b** In the Navigate and Learn toolbar, click the **Define Object Filter** button. The Define Object Filter dialog box opens.



- c** In the Define Object Filter dialog box, select **All object types** and click **OK**.
- 4 Learn all of the objects from the Mercury Tours Web site Flight Finder page.**

In this step, you instruct QuickTest to learn all of the objects in the Mercury Tours Web site Flight Finder page that match your filter and to add them to a shared object repository.

- a** Click the **Find a Flight** page title bar to change the focus from the Navigate and Learn toolbar to the Web page you want QuickTest to learn.
- b** In the Navigate and Learn toolbar, click the **Learn** button. The Web page flickers and the Adding Objects message box is displayed as QuickTest begins adding representations of the objects on the Web page to a new object repository. Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs.

- c In the Navigate and Learn toolbar, click the **Close** button. The Navigate and Learn toolbar closes, and QuickTest and the Object Repository Manager window are visible again.

5 Save the shared object repository.



- a In the Object Repository Manager window, choose **File > Save** or click the **Save** button. The Save Shared Object Repository dialog box opens.
- b Browse to the **Tutorial_ObjectRepositories** folders and select it. In the **File Name** box, enter **MercuryToursFlightFinder** and click **Save**.

6 Create a new shared object repository, add objects to it, and save the repository.

- a In the Flight Finder page, click **Continue**. The Select Flight page opens. You are now ready to add objects from this page.
- b Switch to the Object Repository Manager window and choose **File > New**. A blank shared object repository opens.
- c In the Object Repository Manager window, choose **Object > Navigate and Learn**. Both QuickTest and the Object Repository Manager are hidden.
- d Click the **Select Flight** page title bar to change the focus from the Navigate and Learn toolbar to the Web page you want QuickTest to learn.
- e In the Navigate and Learn toolbar, click the **Learn** button. The Web page flickers and the Adding Objects message box is displayed as QuickTest begins adding representations of the objects on the Web page to a new object repository. Adding these objects takes a few seconds. Do not interact with the Web page while Navigate and Learn runs.
- f In the Navigate and Learn toolbar, click the **Close** button. The Navigate and Learn toolbar closes, and QuickTest and the Object Repository Manager window are visible again.



- g** In the Object Repository Manager window, choose **File > Save** or click the **Save** button. The Save Shared Object Repository dialog box opens.
- h** Browse to the **Tutorial_ObjectRepositories** folder and select it. In the **File Name** box, enter **MercuryToursSelectFlight** and click **Save**.

7 Create new shared object repositories for each of the remaining Web pages.

- a** Repeat step 6 to create shared object repositories for the following pages:

- **Book a Flight**
- **Flight Confirmation**

- b** Name the object repositories **MercuryToursBookFlight** and **MercuryToursFlightConfirmation**, respectively.

8 Associate the **MercuryToursFlightFinder.tsr object repository with the **FlightFinder** action.**



- a** Switch to QuickTest and open the Resources pane, if it is not already open, by choosing **View > Resources** or clicking the **Resources Pane** button.
- b** In the Resources pane, right-click the **FlightFinder** action and choose **Associate Repository with Action**. The Open Shared Object Repository dialog box opens.
- c** Locate and select the **MercuryToursFlightFinder.tsr** shared object repository and click **Open**. The object repository is now associated with the **FlightFinder** action, and is displayed in the Resources pane as a child of that action.

9 Associate the remaining object repositories with the relevant actions.

Repeat step 8 and:

- Associate **MercuryToursSelectFlight.tsr** with the **SelectFlight** action.
- Associate the following repositories with the **BookFlight** action:
 - **MercuryToursBookFlight.tsr**
 - **MercuryToursFlightConfirmation.tsr**

Later, when you add steps to each action, all of the required test objects will be available for use.

10 Save your test.



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

4

Creating Functions and Function Libraries

QuickTest provides built-in functions and methods that will satisfy many of your testing needs. However, there may be times when you need to perform a task for which no suitable method exists. For example, you may want to generate a text file and save it to the file system, or you may want to add a step that accesses data from a Microsoft Excel worksheet. You can create a user-defined function to perform this task, save it in a function library file, and then insert it as a step every time that you need to perform this task.

This lesson includes:

- Introducing Functions and Function Libraries on page 49
- Creating a Function on page 50
- Associating the Function Library with Your Test on page 52

Introducing Functions and Function Libraries

In this lesson, you will use the QuickTest function library window to create a function and associate it with your test. Associating a function library with a test enables you to call any of the functions in the function library from that test.

A **function** is a set of coded steps that perform a particular task. You can create your own user-defined functions to perform operations that are not normally available for use with a particular test object class. In this lesson, you will create a function that checks the date format on a page generated by the Mercury Tours Web site.

Creating a Function

In this lesson, you will create a function that you will later call from your test. This function checks whether a date is displayed in MM/DD/YYYY format. The function also verifies that the date is potentially valid, for example, that the month does not exceed 12 or that the date does not exceed 31.

1 Start QuickTest and create a new function library.

- a If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- b Create a new function library.
 - If the Start Page window opens, in the **Welcome!** area, click **New Function Library**.
 - If QuickTest opens without displaying the Start Page window, choose **File > New > Function Library** or click the **New** button down arrow and choose **Function Library**.



2 Paste the following function into the function library window.

'The following function checks whether a date string (dateStr)
'has 10 characters representing MM/DD/YYYY

```
Function check_data_validity( dateStr )  
  
    Dim firstSlashPos, secondSlashPos  
    Dim mmPart, ddPart, yyyyPart  
    firstSlashPos = inStr( dateStr , "/" )  
    secondSlashPos = inStrRev( dateStr, "/" )  
    If ( firstSlashPos <> 3 or secondSlashPos <> 6 ) Then  
        reporter.ReportEvent micFail,"Format check", "Date string is missing at least one slash ( / )."  
        check_data_validity = False  
        Exit function  
    End If  
  
    mmPart = mid( dateStr, 1,2 )  
    ddPart = mid ( dateStr, firstSlashPos+1, 2 )  
    yyyyPart = mid( dateStr, secondSlashPos +1 , 4 )  
  
    If mmPart > 12 Then  
        reporter.ReportEvent micFail, "Format Check" , "The month value is invalid. It exceeds 12."  
        check_data_validity = False  
        Exit function  
    End If  
    If ddPart > 31 Then  
        reporter.ReportEvent micFail, "Format Check" , "The date value is invalid. It exceeds 31."  
        check_data_validity = False  
        Exit function  
    End If  
    If yyyyPart < 2000 Then  
        reporter.ReportEvent micFail, "Format Check" , "The year value is invalid. (Prior to 2000)"  
        check_data_validity = False  
        Exit function  
    End If  
  
    check_data_validity = True  
  
End Function
```

3 Save the function library.



- a Choose **File > Save** or click the **Save** button. The Save Function Library dialog box opens.
- b Save the function library in the suggested location, naming it **CheckDateFunction.qfl**.

4 Close the function library.

Choose **File > Close**.

Associating the Function Library with Your Test

Now that you created a function and stored it in a function library, you need to associate the function library with your test so that its function will be available to your test. After you associate the function library with your test, you will verify that it was associated successfully.

1 If it is not currently open, open the MercuryTours test you created in Lesson 2, "Creating an Action," in one of the following ways:



- Choose **File > Open > Test**, navigate to the **MercuryTours** test and click the **Open** button.
- Click the **Open** button or the **Open** button down arrow. The Open Test dialog box opens. Navigate to the **MercuryTours** test and click the **Open** button.

2 Associate the CheckDateFunction.qfl with the Mercury Tours test.



- a In QuickTest, open the Resources pane, if it is not already open, by choosing **View > Resources** or clicking the **Resources Pane** button.
- b In the Resources pane, right-click the **Associated Function Libraries** node and choose **Associate Function Library**. The Open Function Library dialog box opens.

- c Locate and select the **CheckDateFunction.qfl** function library and click **Open**. Click **Yes** in the Automatic Relative Path Conversion dialog box. This converts the path to a relative path. (Using a relative path keeps the path valid when you move folders containing tests and other files from one location to another, as long as the folder hierarchy remains the same.)

The function library is now associated with the **Mercury Tours** test, and is displayed in the Resources pane as a child of the **Associated Function Libraries** node.

3 Verify that the function library file is associated with your test.

Choose **Resources > Associate Function Libraries** and verify that **CheckDateFunction.qfl** is listed in the menu.

4 Save the test.



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

Lesson 4 • Creating Functions and Function Libraries

5

Creating a Test

Now that you have created the required automation infrastructure, you are ready to create your tests.

This lesson includes:

- Preparing to Create a Test on page 55
- Adding Steps to the Login Action on page 56
- Analyzing the Login Action in the Keyword View on page 62
- Adding the Remaining Steps to Your Test on page 64
- Dividing an Action into Two Actions on page 78

Preparing to Create a Test

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

- In Microsoft Internet Explorer, clear the **AutoComplete** option for user names and passwords (choose **Tools > Internet Options > Content tab > AutoComplete** Button).
- Close all browsers before you begin.
- Set the proper Web Page/Frame options. Open QuickTest, choose **Tools > Options** and click the **Web > Page/Frame** node. 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.

Adding Steps to the Login Action

A step represents an operation to be performed on your application. For example, a step might click an image or insert a value into an edit box.

In Lesson 2, "Analyzing Your Application," you created unique actions for most of the pages in the Mercury Tours Web site and added them to your test. This enables you to reuse each action in your tests. For example, you may want to use the Login action in multiple tests.

In this section, you will make a reservation for a flight from New York to San Francisco on the Mercury Tours Web site, adding steps in various ways. After you add steps to the first action, you will analyze the steps to gain a better understanding of the Keyword View.



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

- a If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- b Browse to and open the **MercuryTours** test.



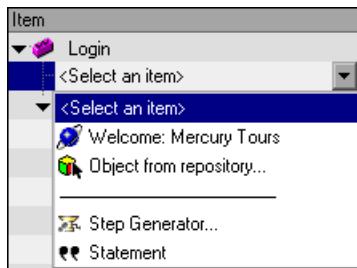
2 Open the Login action in the Keyword View.

- a Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b Make sure that the Keyword View is displayed.
Click the **Keyword View** tab if the Expert View is displayed.
- c In the Test Flow pane, double-click the **Login** action.

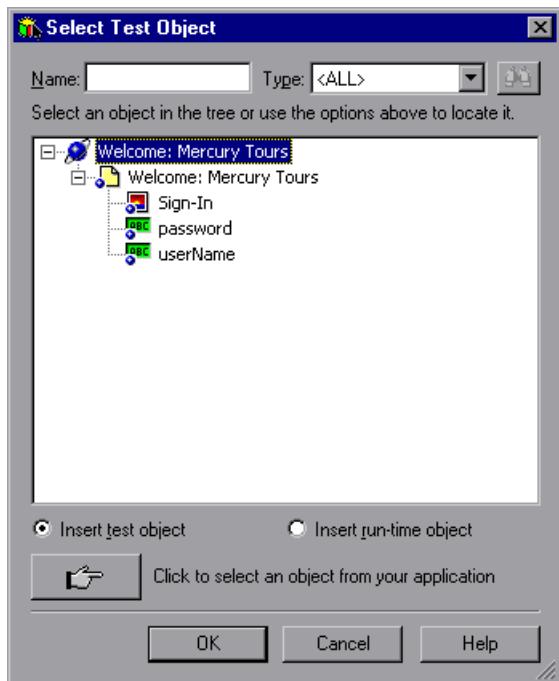
The **Login** action opens.

3 Add the first step needed to log in to the Mercury Tours Web site.

- a In the Keyword View, choose **Insert > New Step**. The **Item** list opens, displaying the top-level (parent) test object in the associated **Login** shared object repository (the **Welcome: Mercury Tours** browser object). You need to insert steps only on the objects on which you want to perform operations, so you do not need to insert a step on this object.



- b Select **Object from repository** to open the Select Test Object dialog box and expand the test object tree.



c Select **userName** and click **OK**.

The Select Object for Step dialog box closes and one step is added to the action, even though three rows are added to the Keyword View. (QuickTest adds a row for each of the parent test objects, but does not perform an operation on these objects. Therefore, these rows are part of the path to the object upon which the step is performed.) During a run session, QuickTest uses the parent objects to help identify the actual object on which it needs to perform an operation.

In this step, the selected **userName** WebEdit test object is added to the **Item** cell; the default method, **Set**, is added to the **Operation** cell; but no documentation is added to the **Documentation** cell because the step is still missing a required value in the **Value** cell.

Item	Operation	Value	Documentation
▼ Login			
▼ Welcome: Mercury Tours			
▼ Welcome: Mercury Tours			
userName	Set		<No documentation summary is available>

d In the **Value** cell, enter **tutorial**.

Inserting this value completes the step. Documentation for the step in plain English is displayed in the **Documentation** cell when you click another area in the Keyword view.

Item	Operation	Value	Documentation
▼ Login			
▼ Welcome: Mercury Tours			
▼ Welcome: Mercury Tours			
userName	Set	"tutorial"	Enter "tutorial" in the "userName" edit box

- e Click the **Expert View** tab to view the syntax of the step in VBScript.

```
Browser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours").WebEdit("userName").Set "tutorial"
```

Notice that this step is being performed on the WebEdit (edit box) test object and that:

- The test object hierarchy for the WebEdit (edit box) test object is displayed prior to the test object.
- The method to be performed on the object is displayed immediately after the test object.
- Full stops (periods) separate each part of the step.

- f Click the **Keyword View** tab to return to the Keyword View.

4 Add the next step.

- a Choose **Insert > New Step** again to insert the next step.

The **Item** list opens listing the sibling objects of the previous step's test object.

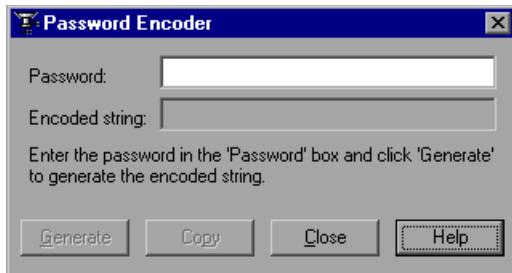
- b Select **password** from the **Item** list.

The **password** WebEdit test object is added to the **Item** cell, and the default method, **Set**, is added to the **Operation** cell. You need to change this method because the password needs to be encoded. Also no documentation is added to the **Documentation** cell because the step is still missing a required value.

- c Click in the **Operation** cell to display the down arrow, and then click the down arrow to display the list of available methods for the selected test object. Select **SetSecure** from the list. This method enables the use of encrypted text. Now you need to generate the encrypted text and insert it in the **Value** cell.

5 Generate an encoded password.

- a On your computer, select **Start > Programs > HP QuickTest Professional > Tools > Password Encoder**. The Password Encoder dialog box opens.



- b In the **Password** box, enter tutorial.
- c Click **Generate**. The Password Encoder encrypts the password and displays it in the **Encoded String** box.
- d Click **Copy**. Paste the encoded value in the **Value** cell for the password step.
Inserting a value completes this step. Documentation for this step is displayed in the **Documentation** cell.
- e Click **Close** to close the Password Encoder.

If you ran your action at this point, QuickTest would automatically open the Mercury Tours Web site and insert the values you specified in the **User Name** and **Password** boxes, as shown below:



6 Insert the last step in the Login action.

- a Click in the **Item** column below the last step to insert the next step.

The **Item** list opens listing the sibling objects of the previous step's test object.

- b Select **Sign-In** from the **Item** list.

This step instructs QuickTest to click the **Sign-In** button.

7 Save your test.



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

Analyzing the Login Action in the Keyword View

Let's take a look at the steps in your first action. Each step represents an operation to be 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, if required. 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 "Sign-In" 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 each action by choosing **View > Expand All**.

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

Item	Operation	Value	Documentation
>Login			
Welcome: Mercury Tours			
Welcome: Mercury Tours			
userNmae	Set	"tutorial"	Enter "tutorial" in the "userNmae" edit box.
password	SetSecure	"474ad3619d961...	Enter the encrypted password in the "password"
Sign-In	Click		Click the "Sign-In" image.

The table below provides an explanation of the steps shown in the action in the Keyword View above.

Step	Description
Login	Login is the action name. Actions are test units that divide your test into logical sections.
Welcome: Mercury Tours	The browser invokes the Welcome: Mercury Tours Web site.
Welcome: Mercury Tours	Welcome: Mercury Tours is the name of the Web page.
userNmae	userNmae is the name of the edit box. Set is the method performed on the edit box. tutorial is the value of the edit box.
password	password is the name of the edit box. SetSecure is an encryption method performed on the edit box. 4bfa6811971c378f7ba56a3c3ebcb467c964fa843426 is the encrypted value of the password.
Sign-In	Sign-In is the name of the image link. Click is the method performed on the image.

Take a few minutes to examine the Keyword View in detail, tracing the steps you performed while creating this action.

For more information on the Keyword View, see the *QuickTest Professional User Guide*.

Adding the Remaining Steps to Your Test

Now that you are familiar with the Keyword View, you can continue adding steps to your test. You will:

- "Add Steps to the FlightFinder Action" on page 64
- "Add a Step to the SelectFlight Action" on page 70
- "Add Steps to the BookFlight Action" on page 72

Add Steps to the FlightFinder Action

In the previous section, you added steps to the **Login** action. If you were to run your test at this point, the last step would open the Flight Finder page in the Mercury Tours Web site.

Now, you will record steps for the **FlightFinder** action you created for the Flight Finder page. The steps you add will use test objects from the associated **MercuryToursFlightFinder** shared object repository.

1 In QuickTest, open the MercuryTours test, if it is not already open.

- a If QuickTest is not already open, choose **Start > Programs > HP QuickTest Professional > QuickTest Professional**.
 - If the Start Page window opens, click **Open Test**.
 - If QuickTest opens without displaying the Start Page window, choose **File > Open > Test** or click the **Open** button down arrow and choose **Test**.
- b In the Open Test dialog box, locate and select the **MercuryTours** test, and then click **Open**.



2 Open the FlightFinder action.

- a** Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b** In the Test Flow pane, double-click the **FlightFinder** action. The FlightFinder action opens.

3 Launch the Mercury Tours Web site and open the Flight Finder page.

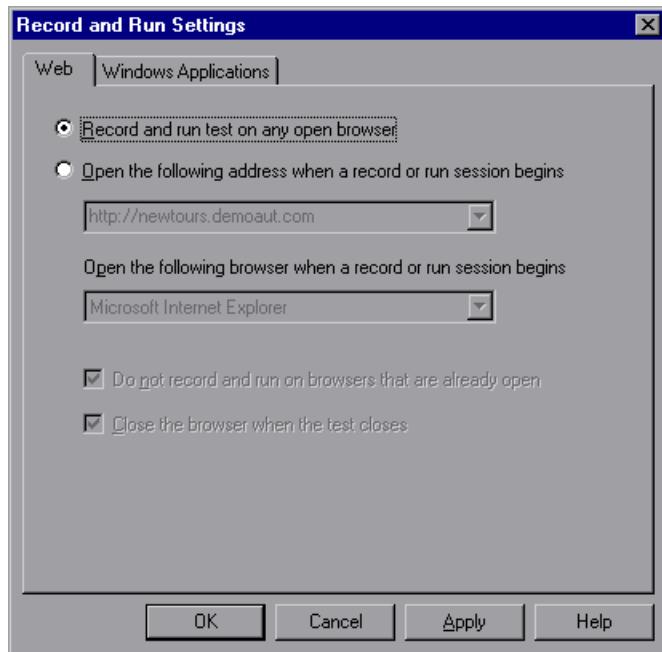
- a** In Microsoft Internet Explorer, enter the following URL:
<http://newtours.demoaut.com>

The Mercury Tours home page opens.

- b** In the Find a Flight area, enter **tutorial** in both the **User Name** and **Password** boxes and click **Sign-In**. The Flight Finder page opens.

The screenshot shows a Windows Internet Explorer window titled "Find a Flight: Mercury Tours". The URL in the address bar is "http://newtours.demoaut.com/mercuryst...". The page features a banner for "one cool summer ARUBA". A navigation menu on the left includes links for Home, Flights, Hotels, Car Rentals, Cruises, Destinations, and Vacations. A "HTML VERSION" link is also present. A sidebar on the left offers "SAVINGS! Rent A Car" with a "CLICK HERE" button. The main content area is titled "FLIGHT FINDER". It contains instructions: "Use our Flight Finder to search for the lowest fare on participating airlines. Once you've booked your flight, don't forget to visit the Mercury Tours Hotel Finder to reserve lodging in your destination city." Below this is a "Flight Details" section with fields for "Type" (radio buttons for "Round Trip" and "One Way"), "Passengers" (dropdown menu set to 1), "Departing From" (dropdown menu set to "Acapulco"), "On" (date dropdown set to June 9), "Arriving In" (dropdown menu set to "Acapulco"), "Returning" (date dropdown set to June 9), and "Service Class" (radio buttons for "Economy class" (selected), "Business class", and "First class"). An "Airline" dropdown menu is set to "No Preference". At the bottom is a "CONTINUE" button.

- c 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**. This enables QuickTest to record on the open Web page.

- d In QuickTest, choose **Automation > Record** or click the **Record** button. QuickTest begins recording, and your browser displays the Flight Finder page in the Mercury Tours Web site. (You can resize QuickTest to display more of the Web page.)

Change the following selections by choosing from the drop-down list.
(Do not use the calendar option.)

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

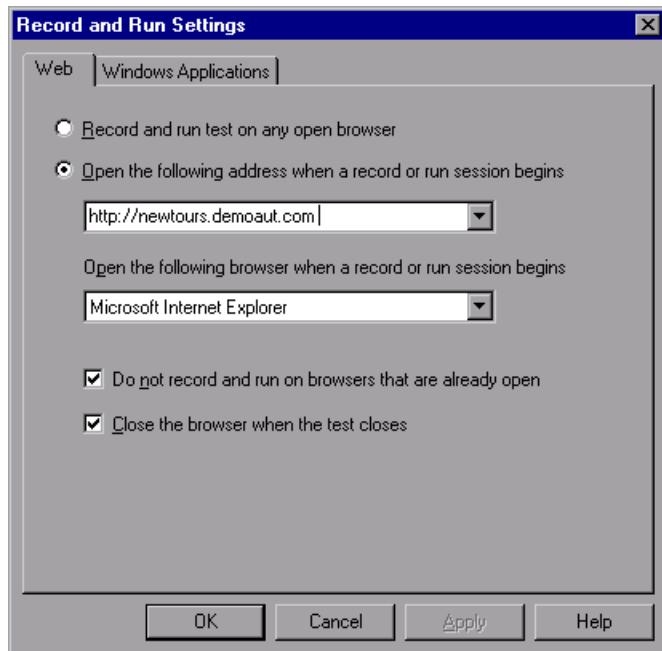
Note: QuickTest records an operation when you make a change in the application. If you accept a default value (or re-select the default value), then QuickTest does not record this as an operation. Therefore, if you are performing this tutorial in the month of December, or on the 29th or 31st of any month, then while recording, select a different month or date, and then select the month or dates listed above, to ensure that the step is recorded. When you stop recording, delete the extra steps with the unnecessary month or date from your test.

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

- e In QuickTest, click **Stop** on the **Automation** toolbar to stop the recording process.

You have now reserved an imaginary 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.

- f In QuickTest, choose **Automation > Record and Run Settings** to restore the required 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 the URL in the first box is <http://newtours.demoaut.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.
- Click **OK** to save your changes and close the dialog box.

4 Save your test.



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

Add a Step to the SelectFlight Action

In the previous section, you added steps to the **FlightFinder** action. If you were to run your test at this point, the last step would open the Select Flight page in the Mercury Tours Web site.

For the purpose of this tutorial, you do not need to make any changes in the Select Flight page, but you do need to instruct QuickTest to accept the default options and continue to the next page. Therefore, in this action, you will add one step that instructs QuickTest to click the **CONTINUE** button.

1 Open the SelectFlight action.



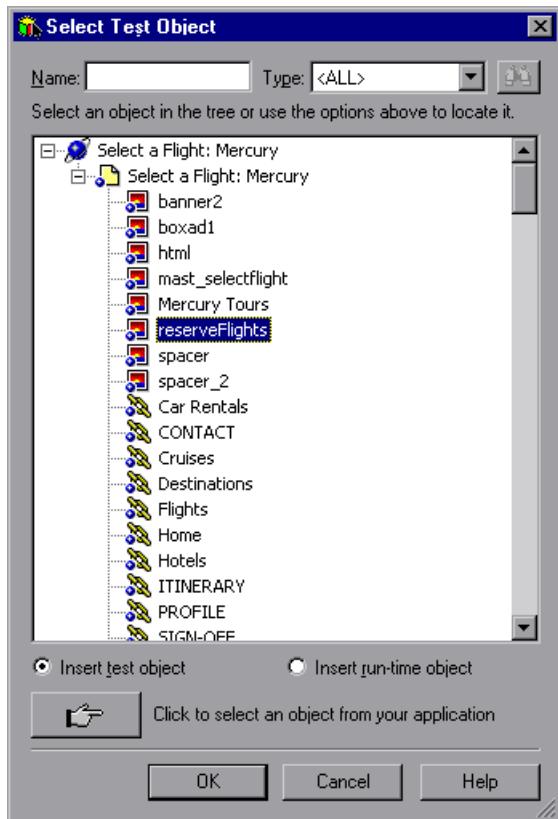
- a Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b In the Test Flow pane, double-click the **SelectFlight** action. The **SelectFlight** action opens.

2 Add a step to the SelectFlight action by clicking in the Item column.

- a Click in the **Item** column below the action name to insert the next step.

The **Item** list opens listing the top-level parent object in the **MercuryToursSelectFlight** shared object repository.

- b** Select **Object from repository** to open the Select Test Object dialog box and expand the test object tree. Notice that this object repository has many more objects than the **MercuryToursLogin** repository because you did not delete extraneous objects from it in Lesson 3, "Creating Object Repositories."



- c** Select **reserveFlights** and click **OK**.

The selected Image test object is added to the step, together with its default method, **Click**. The **Documentation** cell describes what the step does in plain English. Again, this step is displayed on three rows in the Keyword View because the parent test objects are part of the step.

In the Expert View, the step is displayed as follows:

```
Browser("Select a Flight: Mercury").Page("Select a Flight: Mercury").Image("reserveFlights").Click
```

3 Save your test.



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

Add Steps to the BookFlight Action

In the previous section, you added a step to the **SelectFlight** action. If you were to run your test at this point, the last step would open the Book a Flight page in the Mercury Tours Web site.

In this section, you will be creating steps by and pasting content into the Expert View, which is another way of creating steps in your action. The pasted steps do not include values. You will then insert values for each step using the Keyword View.

Finally, you will use the Step Generator to insert a single step. The Step Generator enables you to define an entire step in one dialog box, instead of inserting different parts of a step in the various columns of the Keyword View.

1 Open the BookFlight action.



- a Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b In the Test Flow pane, double-click the **BookFlight** action. The **BookFlight** action opens.

2 Add steps to the BookFlight action.

- a Click the **Expert View** tab to display the Expert View.
- b Copy/paste the following steps:

```
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebEdit("passFirst0").Set  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebEdit("passLast0").Set  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebList("creditCard").Select  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebEdit("creditnumber").Set  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebList("cc_exp_dt_mn").Select  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").WebList("cc_exp_dt_yr").Select  
Browser("Book a Flight: Mercury").Page("Book a Flight: Mercury").Image("buyFlights").Click
```

3 Fill in the missing step details.

- a Click the **Keyword View** tab and look at the **Documentation** column. Notice that for most of the steps, no documentation is available. This is because the steps you just inserted are missing values. For example, in the first step, a value needs to be filled in for the passenger's first name. The details that you need to add are encircled in the following example.

BOOK A FLIGHT

Please review your travel itinerary and make your purchase.

Summary

New York to San Francisco		12/29/2007
FLIGHT	CLASS	PRICE
Blue Skies Airlines 360	Business	270

San Francisco to New York 12/31/2007

FLIGHT	CLASS	PRICE
Blue Skies Airlines 630	Business	270

Passengers: 1
Taxes: \$44
Total Price (including taxes): \$584

Passengers

First Name: Last Name: Meal:

Credit Card

Card Type: Number: Expiration:

- b** In the **Value** cell for the relevant step in the Keyword View, enter the following:

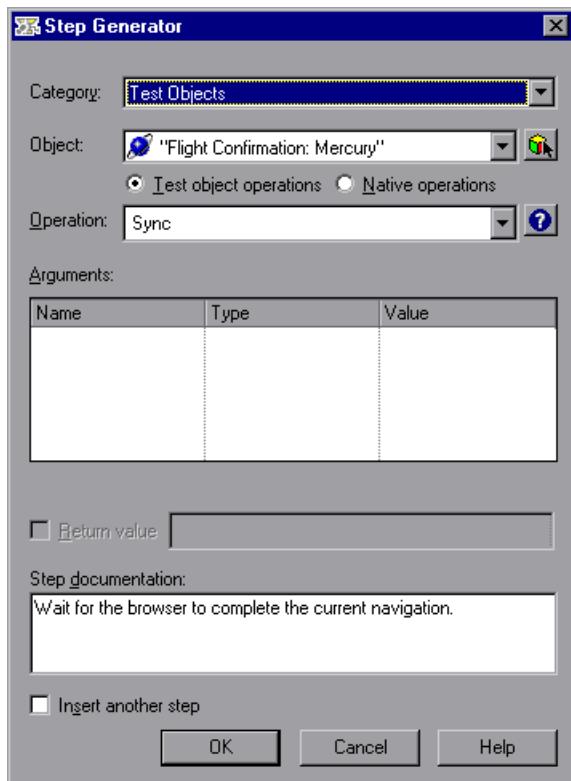
For:	Enter:
passFirst0	Your first name (or a fictitious first name)
passLast0	Your last name (or a fictitious last name)
creditCard	Any of the following credit card names: ➤ American Express ➤ MasterCard (one word) ➤ Visa ➤ Discover ➤ Diners Club ➤ Carte Blanche
creditnumber	Any 8-digit number. (This is a fictitious credit card number.)
cc_exp_dt_mn	Any month from 01 to 12 . Make sure that you enter the value as a two-digit number. (This is the month part of the credit card expiration date.)
cc_exp_dt_yr	Any year from 2008 to 2010 , as a four-digit number. (This is the year part of the credit card expiration date.)

4 Add a step using the Step Generator to confirm the booking and return to the Home page.

The last step in the steps you pasted clicks the **SECURE PURCHASE** button in the Book a Flight page. During a run session, this opens the Flight Confirmation page in the Mercury Tours Web site.

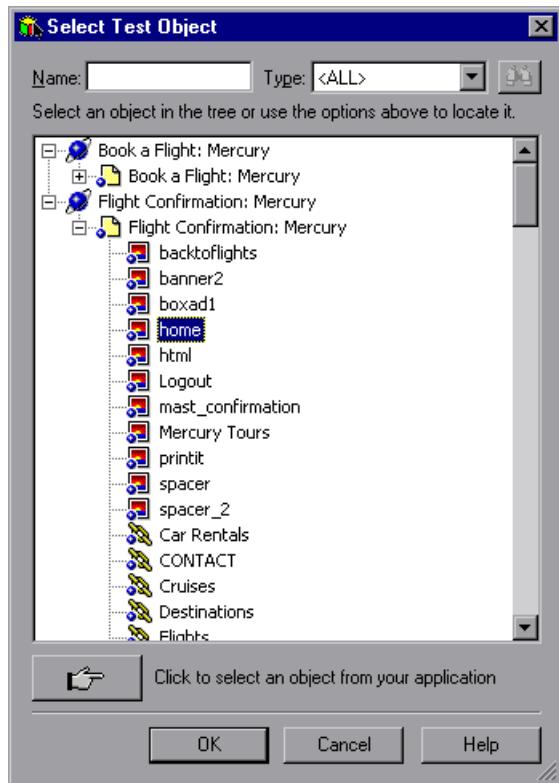
You will now use the Step Generator to define an entire step in one dialog box, instead of inserting different parts of a step in the various columns of the Keyword View, or inserting a step in the Expert View.

- a** Click in the **Item** column below the last step to ensure that the next step will be inserted after the last existing step in this action.
- b** Choose **Insert > Step Generator**. The Step Generator dialog box opens.

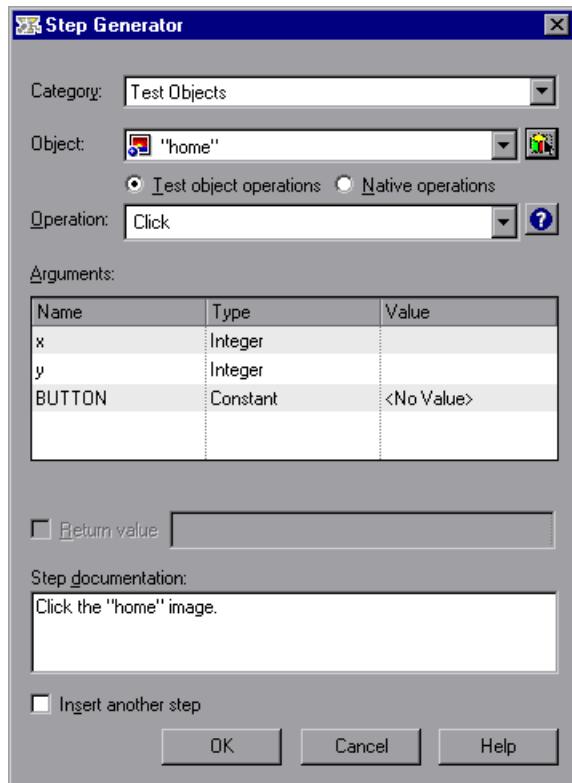




- c Click the **Select Object** button. The Select Test Object dialog box opens. Expand the **Flight Confirmation: Mercury** node of the tree.



- d Select the **home** Image object, and click **OK**. The Step Generator displays the default options for the **home** test object.



You define the arguments and values in the Step Generator just as you would in the Keyword View columns:

- You use the **Category** and **Object** boxes to define the value in the **Item** column in the Keyword View.
- You use the **Operation** cell to define the value in the **Operation** column. In this step, the default operation for the **home** test object, **Click**, is displayed in the **Operation** box. Although you could potentially select a different operation for this test object by clicking the down arrow, **Click** is the operation needed for this step.

- You use the **Arguments** box to define values for the arguments in the **Value** cell in the Keyword View. If any of the arguments were mandatory, a red asterisk would be displayed next to the argument name. In this step, there are no mandatory arguments, and you do not need to define any values.
 - The **Step documentation** box displays the instruction for this step as it will be displayed in the **Documentation** cell of the Keyword View.
 - If you wanted to insert another step, you could select the **Insert another step** check box. (This is the only step you need in this action, so there is no need to select this check box.)
- e** Click **OK**. The Step Generator closes and the step is added to the Keyword View.
- f** Click the **Expert View** tab. You can see that this same step is displayed as follows:

```
Browser("Flight Confirmation: Mercury").Page("Flight Confirmation: Mercury").Image("home").Click
```

5 Save your test.



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

Dividing an Action into Two Actions

Since the BookFlight action currently contains two main processes: purchasing a flight and returning to the Home page, it makes sense to divide the action into separate actions. This will enable you to create different tests that do not return to the Home page after purchasing flights. For example, you may want to create a test that books both flight reservations and hotel reservations, and only then returns to the Home page.

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

- a** In the Keyword View, expand the **BookFlight** action, if needed, by right-clicking the **BookFlight** action row (the top row in the action) and choosing **Expand Sub Tree**.
- b** Select the first **Flight Confirmation: Mercury** row to highlight it.

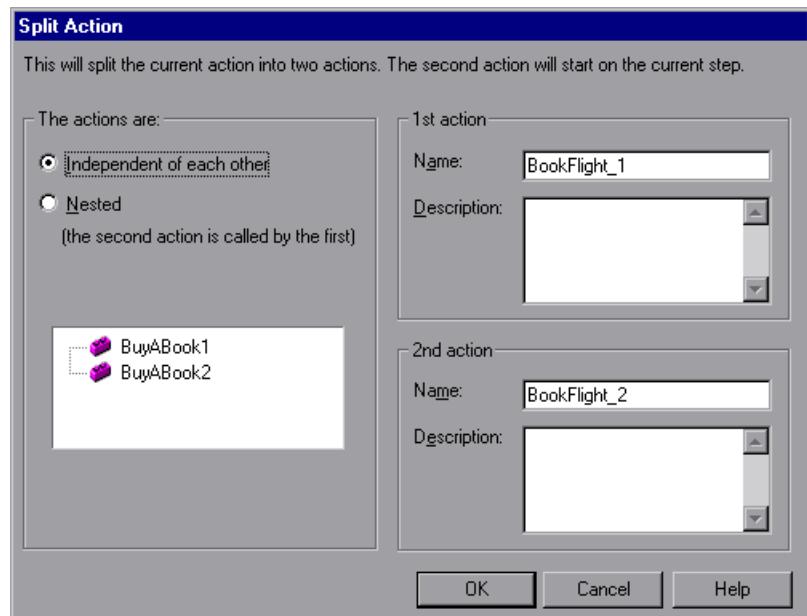
2 Split the test into two actions.



- a** 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 **Flight Confirmation: Mercury** page, which is the currently selected row.
- b** Click **Yes** in the message box. If this reusable action was being called from another test, splitting it into two actions could affect that test adversely because some of the steps would be missing. Since you did not include a call to this action from any other test, you can safely split this action.

The Split Action dialog box opens.

- c** In the Split Action dialog box, enter names and descriptions for the two actions.



- In the **Name** box of the **1st action**, leave the name as **BookFlight**. (Remove **_1** from the action name.)

- In the **Description** box of the **1st action**, enter: Purchases flights.
 - In the **Name** box of the **2nd action** enter: FlightConfirmation
 - In the **Description** box of the **2nd action**, enter: Confirms purchases and returns to the Home page.
- d** Click **OK** to close the Split Action dialog box and accept the other default selections. The two actions are displayed in the Keyword View and in the Test Flow pane.

3 Save your test.



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

4 Close the browser that is open to the Mercury Tours Web site.

Congratulations! You have created your first test and are ready to run it.

6

Running and Analyzing Tests

When you run your test, QuickTest opens the appropriate application and performs each step in the test sequentially. When QuickTest finishes running the test, it displays the results of the run.

This lesson includes:

- Running a Test on page 81
- Analyzing Run Results on page 83

Running a Test

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

1 Start QuickTest and open the MercuryTours test.

- a** If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- b** Browse to and open the **MercuryTours** test.

2 Confirm that all images are saved to the run results.

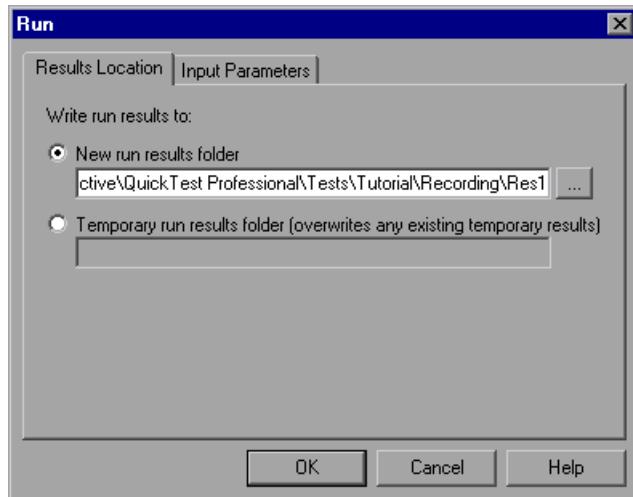
QuickTest allows you to determine when to save images to the run results. In this lesson, all images should be saved to the run results.



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

3 Start running your test.

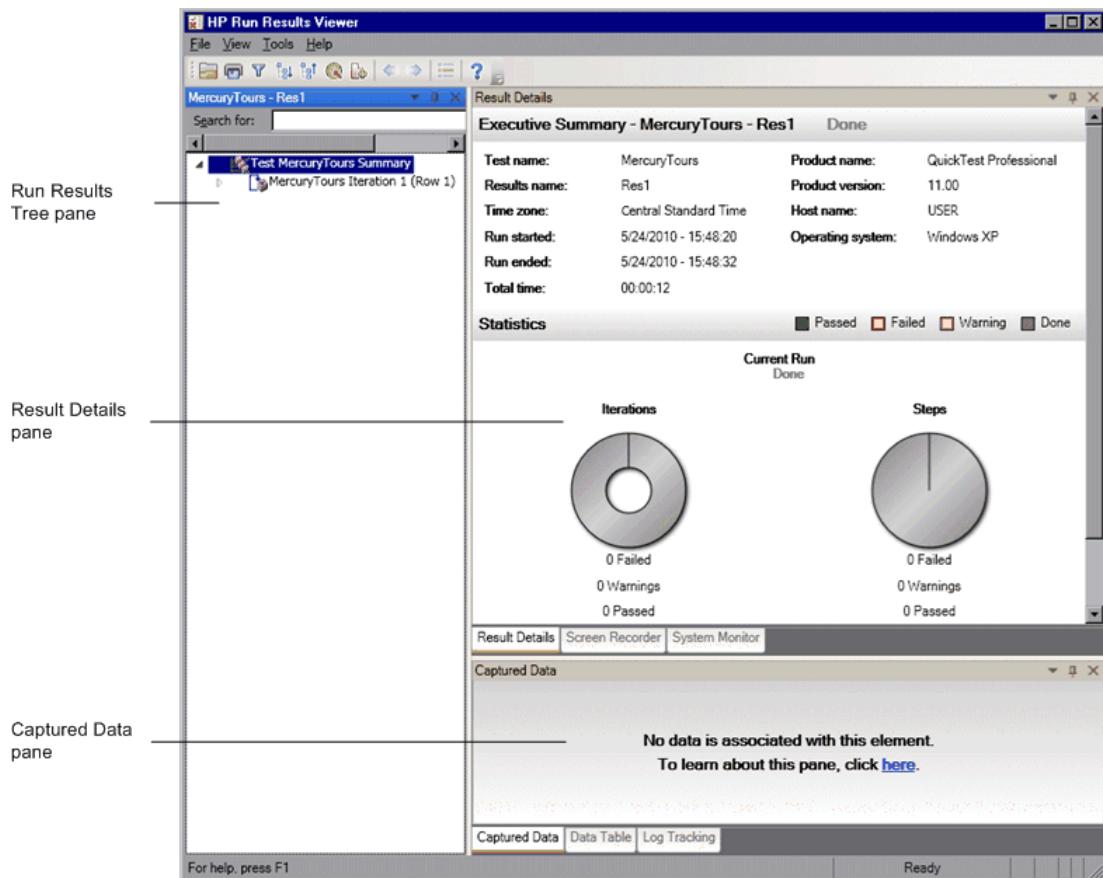
- a** Click **Run** or choose **Automation > Run**. The Run dialog box opens.
- b** Select **New run results folder**. Accept the default results folder name.
- c** Click **OK** to close the Run dialog box.



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

Analyzing Run Results

When QuickTest finishes running the test, the Run Results Viewer opens.



Initially, the Run Results Viewer displays the following panes:

- **Run Results Tree pane.** A graphical representation of the results in an expandable tree, organized according to the actions and the Web pages visited during the test run and can be expanded (**arrow**) to view each step. The steps performed during the 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 run is called an **iteration**, and each iteration is numbered. (The test you ran had only one iteration.)
- **Result Details pane.** A high-level results overview report, containing general information about the test, which steps passed or failed, statistics in pie chart format, and so on.
- **Captured Data pane.** A still image of the state of your application at a particular step.

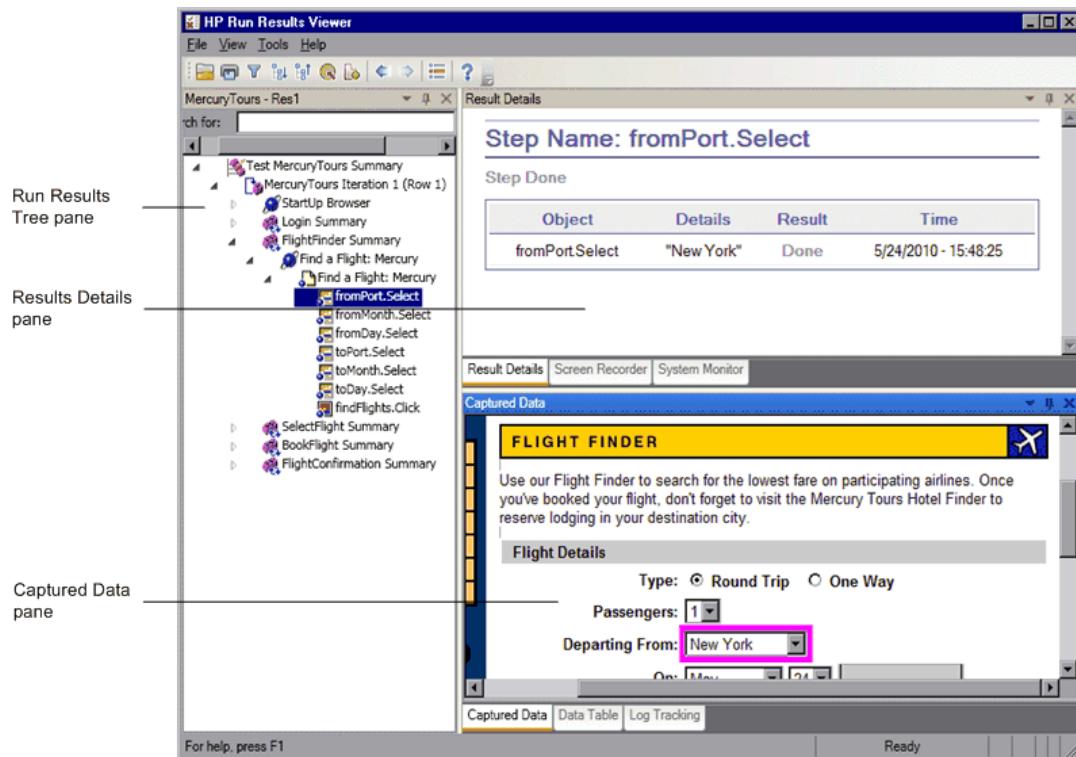
Your test run succeeded because QuickTest was able to navigate the Mercury Tours site according to the steps that you added.

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 results for a specific step.

In the results tree, expand (arrow) **MercuryTours Iteration 1 (Row 1)** > **FlightFinder Summary** > **Find a Flight: Mercury** > **Find a Flight: Mercury** to display all of the steps performed on the Flight Finder page.

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



The Run Results Viewer now contains three panes, displaying:

- The Run Results Tree pane, with one step highlighted
- The Result Details pane, displaying details of the highlighted step
- The Captured Data 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 Run Results Viewer.

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.

7

Creating Checkpoints and Using Functions

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.

In this lesson, you will insert checkpoints and use a function to check the validity of some of the objects in the Mercury Tours Web site.

This lesson includes:

- Understanding Checkpoint Types on page 88
- Checking Objects on page 90
- Checking Pages on page 93
- Checking Tables on page 95
- Checking Text on page 100
- Managing Checkpoints in the Object Repository on page 104
- Running and Analyzing a Test with Checkpoints on page 105
- Performing a Check Using a Function on page 111

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. (This checkpoint type is inserted by selecting the Standard Checkpoint option and then selecting to check a Web Image object.)	Check that the image source file is correct.
Table Checkpoint	Checks information in a table. (This checkpoint type is inserted by selecting the Standard Checkpoint option and then selecting to check any table object.)	Check that the value in a table cell is correct.
Page Checkpoint	Checks the characteristics of a Web page. (This checkpoint type is inserted by selecting the Standard Checkpoint option and then selecting to check a Web Page object.)	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 an 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 an application after capturing it as a bitmap.	Check that a Web page (or any portion of it) is displayed as expected.

Checkpoint Type	Description	Example of Use
Database Checkpoint	Checks the contents of databases accessed by an application or Web site.	Check that the value in a database query is correct.
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.	<p>Check the content of an element to make sure that its tags, attributes, and values have not changed.</p> <p>Note: 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.</p>

You can add most checkpoints to your test either while inserting steps or while recording. The following sections explain how to create some of the checkpoints described above, in the test you created in "Adding Steps to the Login Action" on page 56.

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, see the *HP QuickTest Professional User 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.

To insert the checkpoint, the Mercury Tours application needs to be open to the page on which you are inserting the checkpoint.

1 Start QuickTest and open the MercuryTours test.

- a** If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- b** Browse to and open the **MercuryTours** test.

2 Save the test as Checkpoint.

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

3 Display the action step on which 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.



- a** Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b** In the Test Flow pane, double-click the **BookFlight** action.

4 Open the Mercury Tours application to the Book a Flight page.

- a** Log in to the Mercury Tours Web site: <http://newtours.demoaut.com>

In the **User Name** and **Password** boxes, type **tutorial**.

Click **Sign-In**. The Flight Finder page opens.

- b** Enter flight details.

Change the following selections by choosing from the drop-down list.
(Do not use the calendar option.)

- Departing From: **New York**
- On: **December 29**

► Arriving In: **San Francisco**

► Returning: **December 31**

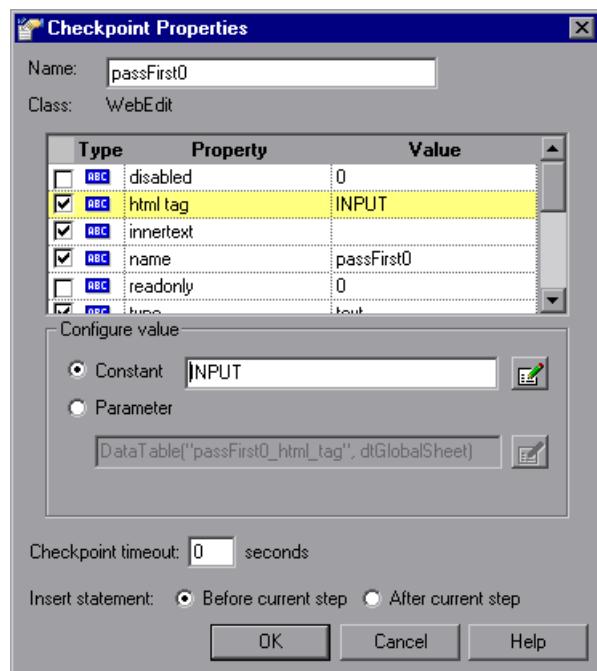
Click **CONTINUE** to accept the other default selections. The Select Flight page opens.

- c In the Select Flight page, click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

5 Create a standard checkpoint.



- a In the **Item** column of the Keyword View, highlight the **passFirst0** row.
- b Choose **Insert > Checkpoint > Standard Checkpoint**. 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	INPUT 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	passFirst0 is the name of the edit box.
type	text	text is the type of object as defined in the HTML source code.
value		Currently, the value is empty. You need to enter the same value that you specified for the First Name edit box.

- c In the **Name** box, enter **CheckName** as the new checkpoint name.
- d Scroll down in the object properties area and select the row containing the property named **value**. You need to enter the value that matches the value you entered in the actual step. To do this, click in the **Constant** box and enter the first name value.
- e In the **Insert statement** area of the Checkpoint Properties dialog box, select **After current step**. This inserts the checkpoint after the **passFirst0 Set...** step. (This is the step in which the passenger's first name is entered in the **First Name** box.)

- f 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"
	passFirst0	Check	CheckPoint("CheckName")
	passLast0	Set	"Jones"

Enter "Nicole" in the "passFirst0" edit box.
Check whether the "passFirst0" edit box has the proper
Enter "Jones" in the "passLast0" edit box.

6 Save the test.



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

You can insert many 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 QuickTest learned the objects on which you inserted steps in your test.

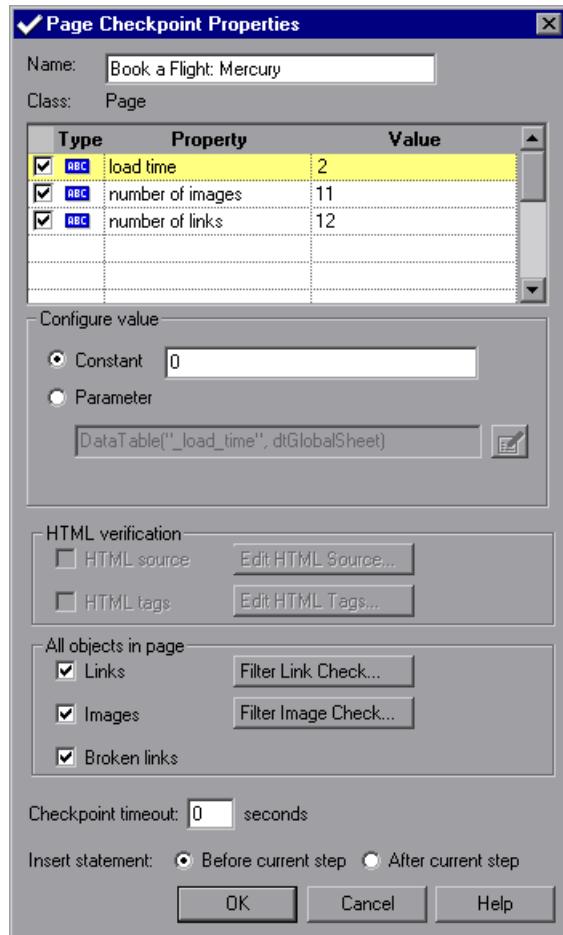
1 Locate the page where you want to add a page checkpoint.



In the Test Flow pane, double-click the **BookFlight** action, if the action is not already open. In the **Item** column of the Keyword View, highlight the **Book a Flight: Mercury** row. (This is the row directly above the row displaying **passFirst0**.)

2 Create a page checkpoint.

- a Choose **Insert > Checkpoint > Standard Checkpoint**. 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. The load time of the page may vary from computer to computer and your results may not match the above image.

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

- b** In the **Name** box, enter **CheckLinks** as the new checkpoint name.
- c** If all three check boxes are not selected in the **Type** column, select them.
- d** 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.



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.

- a** In the Test Flow pane, double-click the **BookFlight** action, if the action is not already open. The action is displayed in the Keyword View.
- b** Highlight the **passFirst0** step (the step that enters the passenger's first name into the **First Name** edit box).

2 Open the Mercury Tours Web site to the Book A Flight page.

- a** In the **User Name** and **Password** boxes of the Mercury Tours Web site (<http://newtours.demoaut.com>), type **tutorial**. (You opened the Mercury Tours Web site in step 4 on page 90.)

Click **Sign-In**. The Flight Finder page opens.

b Enter flight details.

Change the following selections by choosing from the drop-down list.
(Do not use the calendar option.)

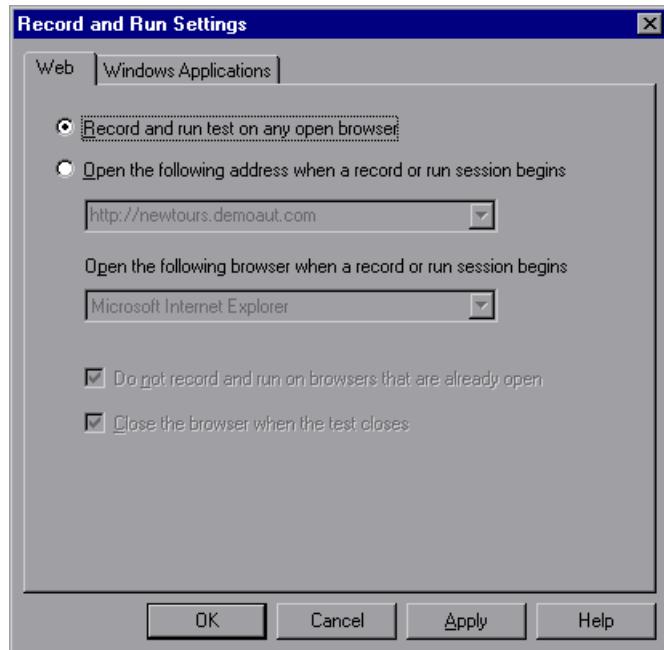
- Departing From: **New York**
- On: **December 29**
- Arriving In: **San Francisco**
- Returning: **December 31**

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

- c** In the Select Flight page, click **CONTINUE** to accept the default flight selections. The Book a Flight page opens.

3 Configure QuickTest to record on the open browser page.

- a** In QuickTest, choose **Automation > Record and Run Settings**. The Record and Run Settings dialog box opens.



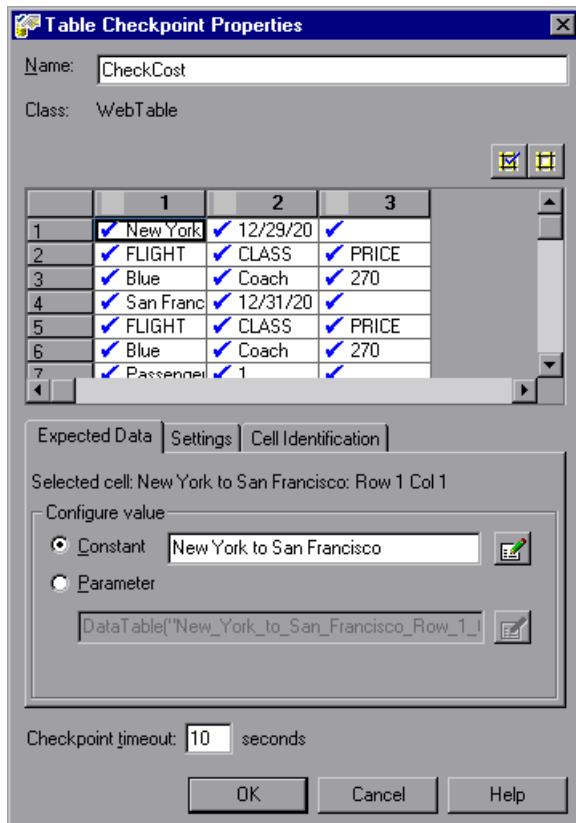
- b** Select the **Record and run test on any open browser** radio button and click **OK** to close the dialog box.

4 Create a table checkpoint.

- a** In the Book Flight page of the Mercury Tours Web site, under the flight departure date and **Price**, highlight **270** (the price of the fare of the first leg between New York and San Francisco).
- b** In QuickTest, click the **Record** button or choose **Automation > Record**. QuickTest begins a recording session.
- c** Choose **Insert > Checkpoint > Standard Checkpoint**. The QuickTest window is hidden, and the pointer changes into a pointing hand.
- d** Click the highlighted string, **270**. The Object Selection dialog box opens. For more information, see the Object Selection Dialog Box section in the *HP QuickTest Professional User Guide*.
- e** Select **WebTable: New York to San Francisco** and click **OK**.

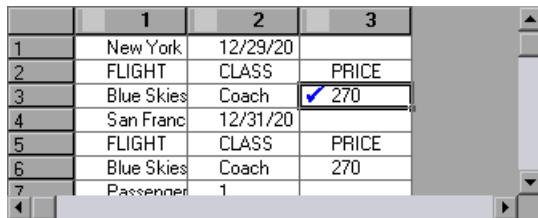
The Table Checkpoint Properties dialog box opens, showing the rows and columns of the table.

- f Enter **CheckCost** as the new checkpoint name in the **Name** box of the Table Checkpoint Properties dialog box.



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.

- g** 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.

- h** 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.



▼ BookFlight			
▼ Book a Flight: Mercury			
▼ Book a Flight: Mercury	Check	CheckPoint("...")	Check whether the "Bool
passFirst0	Set	"Nicole"	Enter "Nicole" in the "pas
Card Type:	Check	CheckPoint("...")	Check whether the conte
passFirst0	Check	CheckPoint("...")	Check whether the "pass

5 Stop the recording session.

Click the **Stop** button to stop recording.

6 Save the test.



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

7 Leave the Mercury Tours Web site open.

In the next section, you will need to navigate to the next page in the Mercury Tours Web site—the Flight Confirmation page. Therefore, you may want to keep the Mercury Tours Web site open to the current page.

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.

- a** In the Test Flow pane, double-click the **FlightConfirmation** action. The action is displayed in the Keyword View.
- b** In the Keyword View, highlight the first row displaying **Flight Confirmation: Mercury**.

2 Open the Mercury Tours Web site to the Flight Confirmation page.

Note: If the Mercury Tours Web site is already open to the Book a Flight page, skip to step d, below.

- a** Log in to the Mercury Tours Web site: <http://newtours.demoaut.com>
In the **User Name** and **Password** boxes, type **tutorial**.
Click **Sign-In**. The Flight Finder page opens.
- b** Enter flight details.
Change the following selections by choosing from the drop-down list.
(Do not use the calendar option.)
 - Departing From: **New York**
 - On: **December 29**

► Arriving In: **San Francisco**

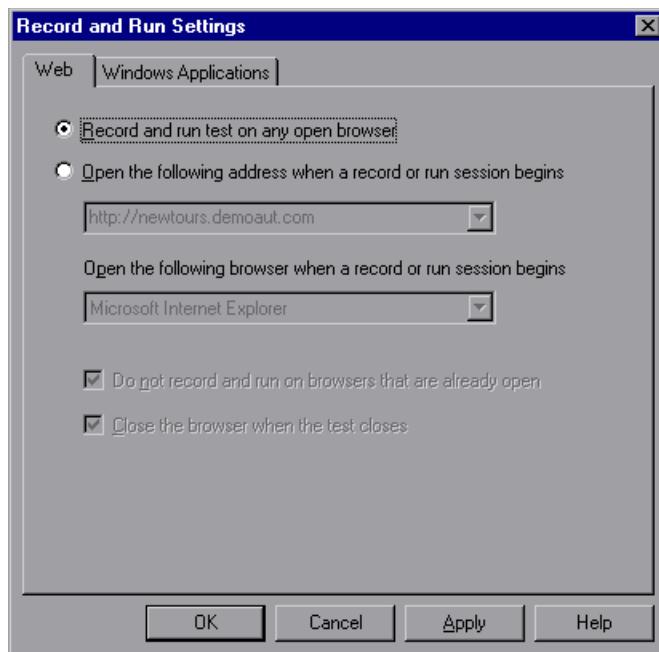
► Returning: **December 31**

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

- c In the Select Flight page, click **CONTINUE** to accept the default flight selections.
- d In the Book a Flight page, click **SECURE PURCHASE**. The Flight Confirmation page opens.

3 Configure QuickTest to record on the open browser page.

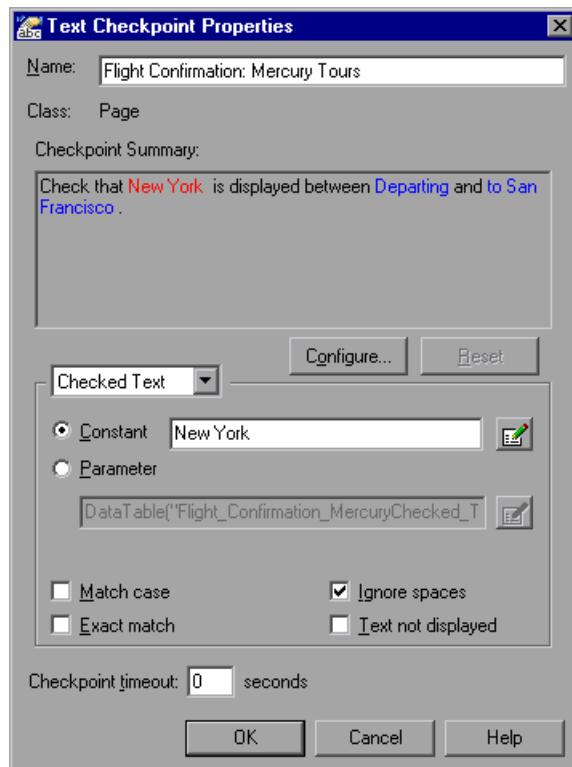
- a In QuickTest, choose **Automation > Record and Run Settings**. The Record and Run Settings dialog box opens.



- b Confirm that the **Record and run test on any open browser** radio button is selected and click **OK** to close the dialog box.

4 Create a text checkpoint.

- a In the Flight Confirmation page, under **Departing**, highlight the text **New York** (include the space after it).
- b Click the **Record** button or choose **Automation > Record**. QuickTest begins a recording session.
- c Choose **Insert > Checkpoint > Text Checkpoint**. The QuickTest window is hidden, and the pointer changes into a pointing hand.
- d Click the highlighted text string, **New York**. 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.
- Remove **San Francisco** from the **Checkpoint Summary** pane, as follows:
Click the **Configure** button to open the Configure Text Selection dialog box. Highlight the **to** text string. (Do not highlight **San Francisco**). Click the **Text After** button. **San Francisco** changes from black to gray. Click **OK** to close the dialog box. San Francisco is removed from the **text after** part of the Checkpoint Summary pane.



- 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: Mercur

5 Stop the recording session.

Click the **Stop** button to stop recording.

6 Make sure that the checkpoint is above the home step.

Drag the step up a row, if needed. Otherwise, during the test run, QuickTest will click the **BACK TO HOME** button before performing the checkpoint.

7 Save the test.



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

Managing Checkpoints in the Object Repository

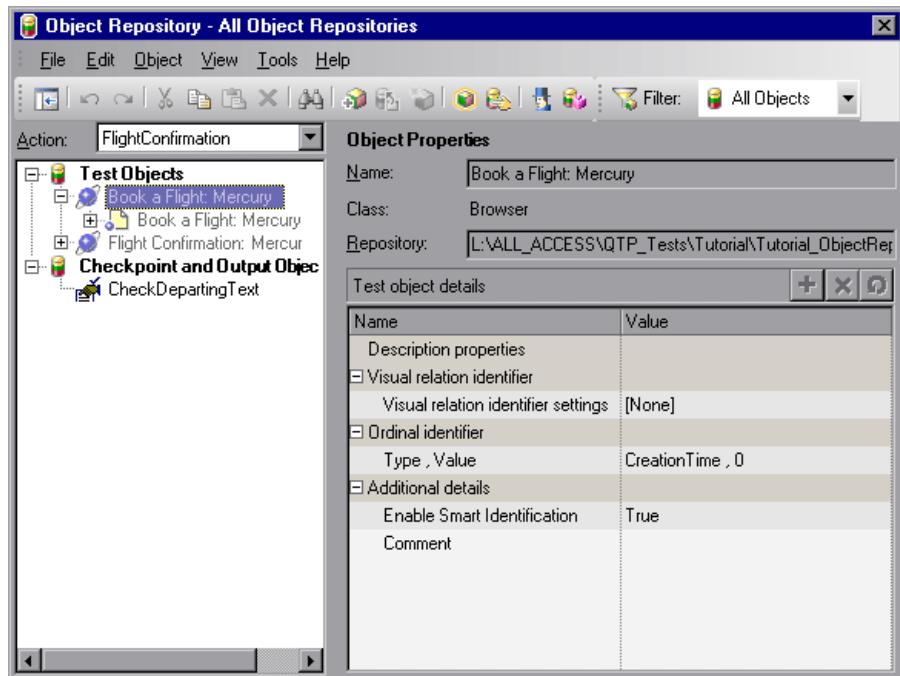
In addition to viewing checkpoints in a specific action, you can also view them in the object repository and modify their properties. If applicable, you can even use the same checkpoint in more than one location. For example, if you want to verify that your organization's logo appears on every page in your application, you can create a checkpoint and insert it in wherever steps open another page. For the purposes of this tutorial, you will not be reusing checkpoints.

1 Open the Object Repository window.



In QuickTest, choose **Resources > Object Repository** or click the **Object Repository** button.

The Object Repository window opens, displaying a tree of all test objects and all checkpoint and output objects in the current action. The tree includes all local objects and all objects in any shared object repositories associated with the action.



Name	Value
Description properties	
Visual relation identifier	
Visual relation identifier settings	[None]
Ordinal identifier	
Type . Value	CreationTime . 0
Additional details	
Enable Smart Identification	True
Comment	

2 Select an Action to view its checkpoints.

In the Action drop-down list, click the down arrow and select an action. You can see the action's test objects, checkpoint objects, and output value objects.

3 Close the Object Repository window.

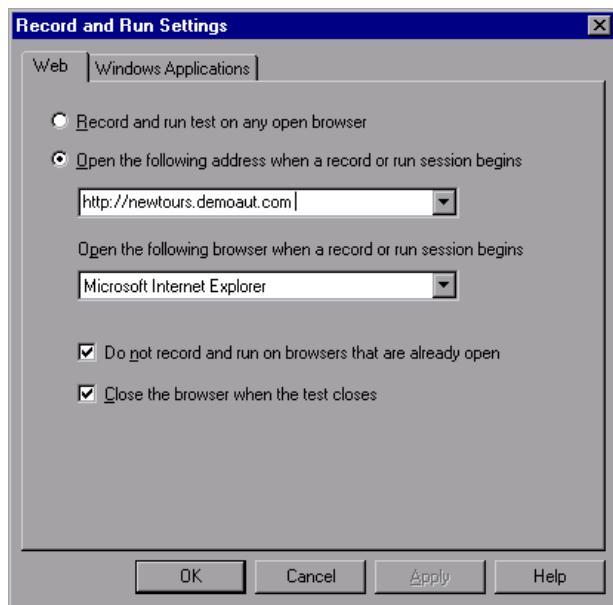
Click the X in the top-right corner or select **File > Close**.

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 Configure QuickTest to open the Mercury Tours Web page.

- a In QuickTest, choose **Automation > Record and Run Settings**. The Record and Run Settings dialog box opens.



- b Select the **Open the following address when a record or run session begins** radio button and click **OK** to close the dialog box.

2 Start running your test.

- a** Click **Run** or choose **Automation > Run**. The Run dialog box opens.
- b** Ensure that **New run results folder** is selected. Accept the default results folder name.
- c** Click **OK**. When the test run is completed, the Run Results Viewer opens.

3 View the run results.

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

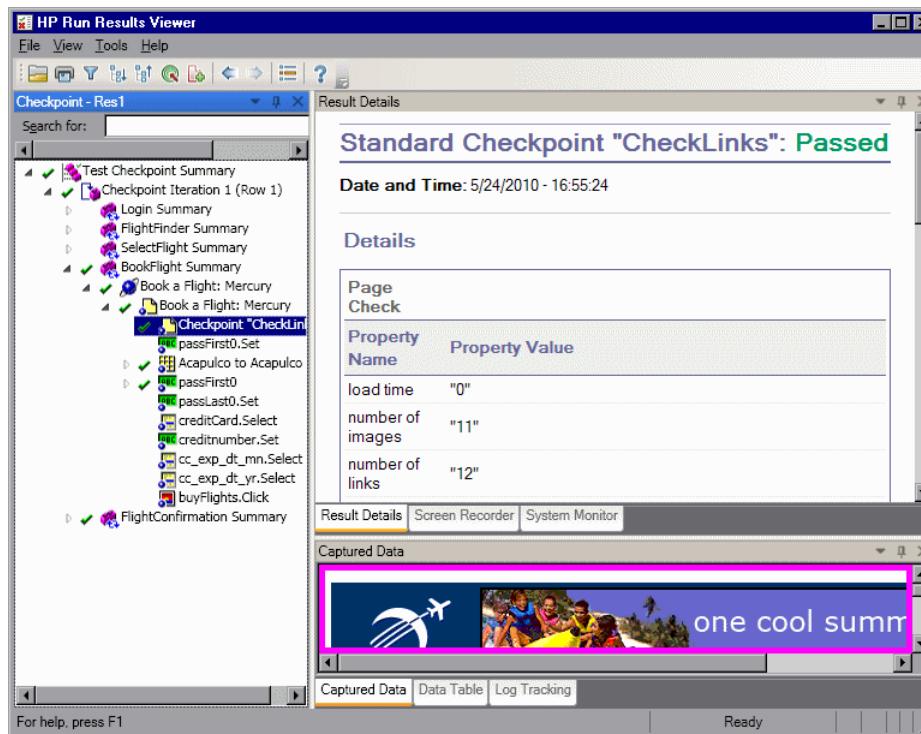
4 View the results of the page checkpoint.

In the run results tree, expand all the child nodes of **Checkpoint Iteration 1 (Row 1)** until you reach **Book a Flight: Mercury**, and highlight **Checkpoint "CheckLinks"**.

In the Result Details pane, you can review the details of the page checkpoint, which lists the items checked. The load time of the page may vary from computer to computer and your results may not match the below image.

Tip: You can drag the bottom of the Result 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.



5 View the results of the table checkpoint.

In the results tree, **Book a Flight: Mercury** page, expand (arrow) **Card Type**, and highlight **Checkpoint "CheckCost"**.

In the Result 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: 11/20/2007 - 0:19:34			
Details			
Verification type: String Content. Settings: Exact match - ON; Ignore space - ON; Match case - OFF. Results: Checked 1 cells; Succeeded: 1; Failed: 0			
1	New York	12/29/20	
2	FLIGHT	CLASS	PRICE
3	Blue Skies	Business	270
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 (arrow) **passFirst0**.

Highlight **Checkpoint "CheckName"**.

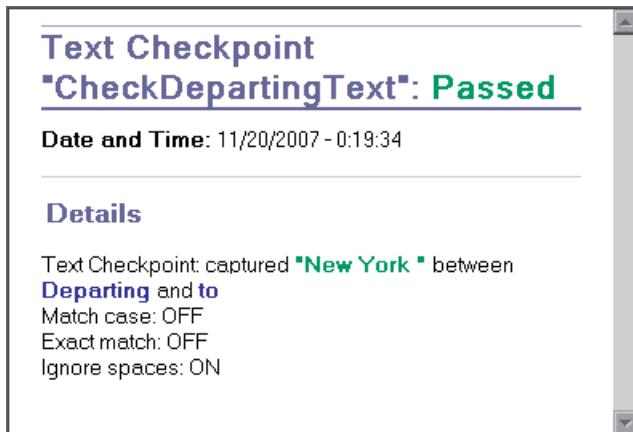
In the Result Details pane, you can review the details of the standard checkpoint, which lists the properties that were checked and their values. The checkpoint passed because the actual values of the checked properties match the expected values.

Standard Checkpoint "CheckName": Passed													
Date and Time: 11/20/2007 - 0:19:34													
Details													
CheckName Results													
<table><thead><tr><th>Property Name</th><th>Property Value</th></tr></thead><tbody><tr><td>html tag</td><td>INPUT</td></tr><tr><td>innertext</td><td></td></tr><tr><td>name</td><td>passFirst0</td></tr><tr><td>type</td><td>text</td></tr><tr><td>value</td><td>Nicole</td></tr></tbody></table>		Property Name	Property Value	html tag	INPUT	innertext		name	passFirst0	type	text	value	Nicole
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 **Flight Confirmation: Mercury** and highlight **Checkpoint "CheckDepartingText"**.

In the Result 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 Run Results Viewer.

Choose **File > Exit**.

Performing a Check Using a Function

In the previous sections, you used checkpoints to check various objects in the Mercury Tours Web site. Now you will use the function you created in Lesson 4, "Creating Functions and Function Libraries" to check the date format of a Web element in the **SelectFlight** action.

1 Start QuickTest and open the Checkpoint test.

- a** If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- a** Browse to and open the **Checkpoint** test.

2 Save the test as Function.

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

3 Display the Select Flight page in The Mercury Tours Web site.

- a** Log in to the Mercury Tours Web site: <http://newtours.demoaut.com>

In the **User Name** and **Password** boxes, type **tutorial**.

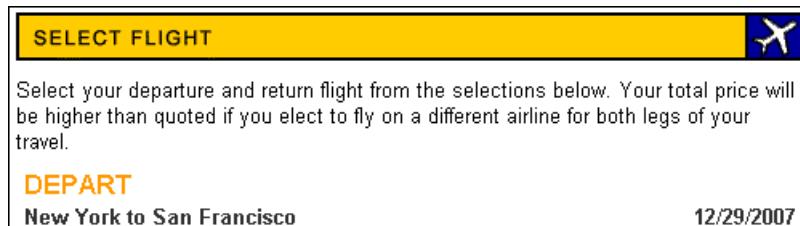
Click **Sign-In**. The Flight Finder page opens.

- b** Enter flight details.

Change the following selections by choosing from the drop-down list.
(Do not use the calendar option.)

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

Click **CONTINUE** to accept the other default selections. The Select Flight page opens. Notice the date displayed in the DEPART area. The date is in MM/DD/YYYY format.



4 Add the 12/29/<current year> object to your repository.

- a** Open the Object Repository Manager (**Resources > Object Repository Manager**).
- b** Browse to and open **MercuryToursSelectFlight.tsr**.
- c** Enable the repository for editing (**File > Enable Editing**).
- d** Click the browser window to make the Mercury Tours Web site active.
- e** Learn the **12/29/<current year>** object.
 - From the Object Repository Manager, select **Object > Add Objects**. The Object Repository Manager window is minimized, and the cursor changes to a pointing hand.
 - Click the text string **12/29/<current year>**.
 - Make sure that the object **WebElement: 12/29/<current year>** is highlighted in the Object Selection dialog box.
 - Click **OK**. The **12/29/<current year>** object is added to your repository.
- f** Save the repository.

5 In QuickTest, display the action in which you want to add a function.

You want to add a function that checks the **innertext** property values of the date displayed in the **SelectFlight** action.

In the Test Flow pane, double-click the **SelectFlight** action.

6 Paste the following in the Expert View.

Click the **Expert View** tab and paste the function shown below before the existing step. (You want the function to run before QuickTest navigates to the next page in the application.)

Note: In the first line of the function, make sure to insert the current year. For example, if you are performing this tutorial in 2011, change 2010 to 2011. In the PDF version of this tutorial, the first line of the code breaks across two lines. If you are copying and pasting from the PDF, make sure to remove the line break after **29/**: so that the first line of the code begins with **departureDate** and ends with **("innertext")**. (The **Page** part should be displayed as: **Page("Select a Flight: Mercury")** with a single space between **Flight:** and **Mercury**.)

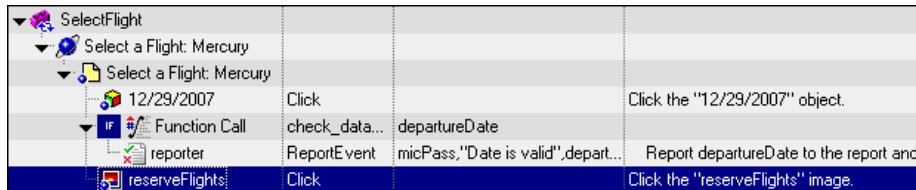
```
departureDate=Browser("Select a Flight: Mercury").Page("Select a Flight: Mercury").WebElement("12/29/2010").GetROProperty("innertext")

if check_data_validity( departureDate ) then
    reporter.ReportEvent micPass, "Date is valid" , departureDate
end if
```

These steps use the **check_data_validity** function defined in the associated function library. You created this function library in Lesson 4, "Creating Functions and Function Libraries."

7 Look at these steps in the Keyword View.

Click the **Keyword View** tab. (The highlighted step is not part of the function—it is the step that navigates to the next page in the Mercury Tours Web site.)



8 Save your test.



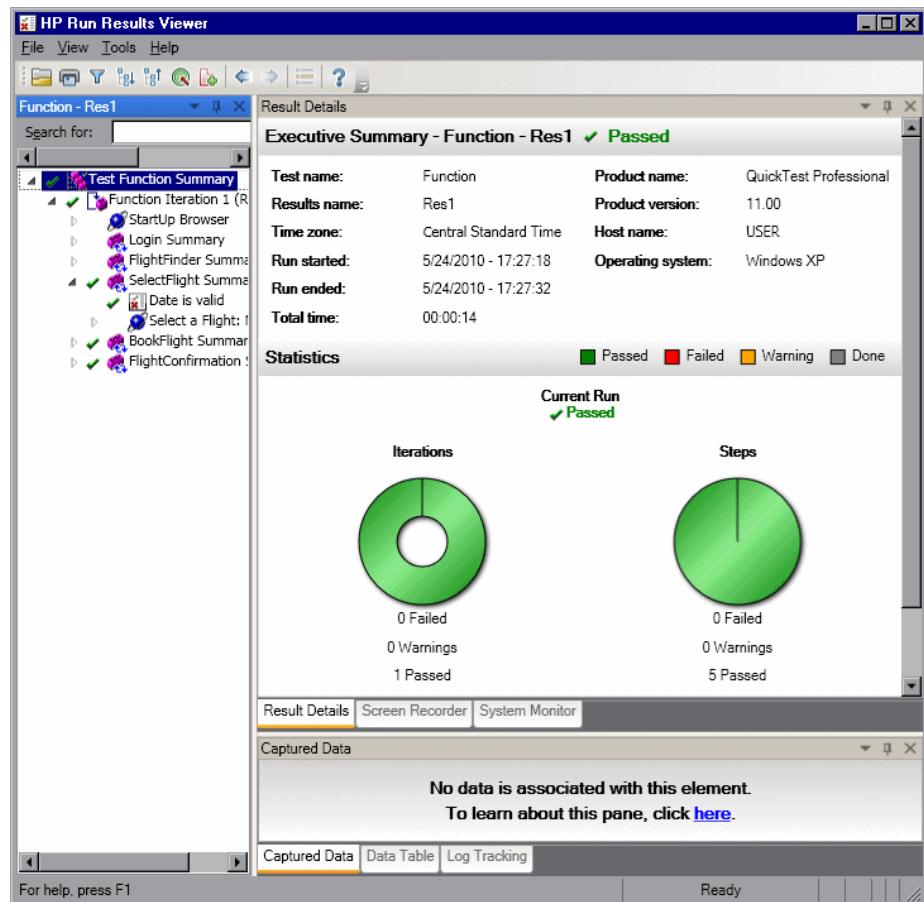
Choose **File > Save** or click the **Save** button.

9 Start running your test.

- a Click **Run** or choose **Automation > Run**. The Run dialog box opens.
- b Ensure that **New run results folder** is selected. Accept the default results folder name.
- c Click **OK**. When the test run is completed, the Run Results Viewer opens.

10 View the run results.

The run result should be **Passed**, indicating that the function passed because the date is displayed in the expected format, and it contains acceptable values.



8

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 create 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.

This lesson includes:

- Defining a Data Table Parameter on page 117
- Adding Parameter Values to a Data Table on page 120
- Modifying Steps Affected by Parameterization on page 121
- Parameterizing an Action on page 123
- Running and Analyzing a Parameterized Test on page 127

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.

- a** If QuickTest is not currently open, open it as described in step 1 on page 29, making sure that only the Web Add-in is loaded.
- b** Browse to and open the **Checkpoint** test.

2 Save the test as Parameter.

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

3 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**.



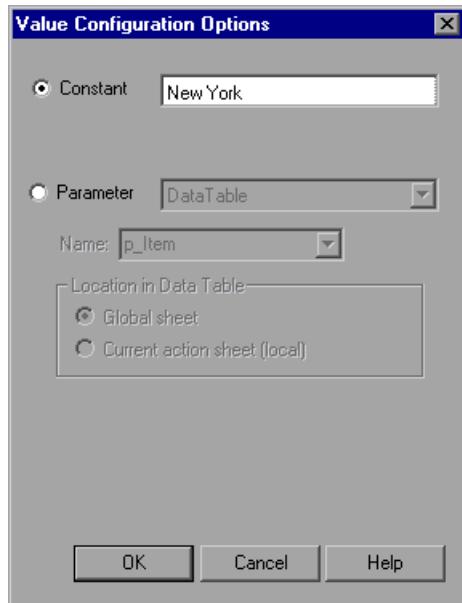
4 Open the FlightFinder action.

- a Click the **Test Flow** tab to display the Test Flow pane. If you do not see the Test Flow tab, choose **View > Test Flow** or click the **Test Flow Pane** button to display it.
- b In the Test Flow pane, double-click the **FlightFinder** action. The FlightFinder action opens.

5 Select the text to parameterize.

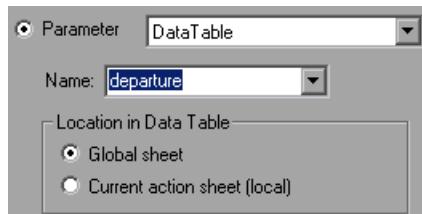
In the **fromPort** row in the Keyword View, click the **Value** cell and then click the parameterization icon .

The Value Configuration Options dialog box opens.



6 Set the parameterization properties.

- a Select the **Parameter** radio button. This lets you replace the constant value (**New York**) with a parameter.
- b 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 interface. The main window displays the 'Keyword View' tab, showing a hierarchical tree structure. Under 'FlightFinder', 'Find a Flight: Mercury' is expanded, revealing four items: 'fromPort', 'fromMonth', 'fromDay', and 'toPort'. The 'fromPort' item has a tooltip: 'Select <departure> Select the "New York" item'. Below this, the 'Data Table' is visible, showing a single row with the header 'A1 New York'. The first cell of the row, corresponding to 'fromPort', contains the value 'New York'. A callout box with the text 'New Column in Data Table' points to this cell.

A1	New York
departure	
1	New York
2	
3	

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 another area in the Keyword view, the step is displayed as:



Adding Parameter Values to a Data Table

As you saw previously, QuickTest displays parameter values in the Data Table. In this section, you will add another departure city to the Data Table (in the FlightFinder action of the Parameter test), so that QuickTest can test the application with this data.

1 Enter an additional city in the departure column.

- a Click row 2 in the **departure** column and enter London.
- b Press **Enter**.

2 Save the test.



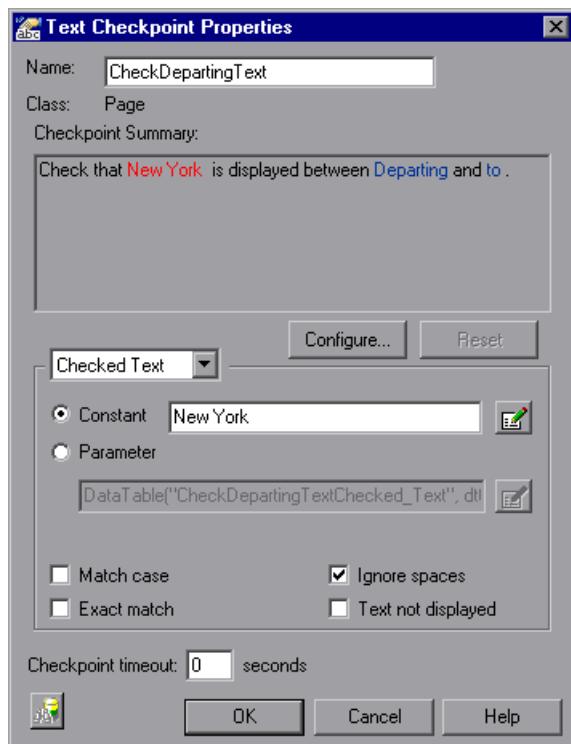
Choose **File > Save** or click the **Save** button.

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 **FlightConfirmation** action, right-click **Flight Confirmation: Mercury** row with the existing checkpoint and select **CHECKPOINT PROPERTIES**. The Text Checkpoint Properties dialog box opens.

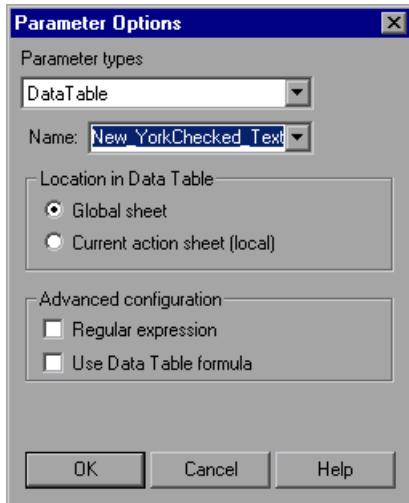


2 Parameterize the text checkpoint.

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.



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



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

Parameterizing an Action

QuickTest enables you to insert various types of parameters. This tutorial introduces you to Data Table parameters. For information on other types of parameters, see the *HP QuickTest Professional User Guide*.

If you look at the Data Table at the bottom of the QuickTest window in the any test, you will see a tab for each action plus a **Global** tab.



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 (five full iterations). In addition, you can create data sets for each action, using the relevant action sheet. If you parameterize a step using a Data Table parameter from a local action sheet and enter five rows of data in that sheet, you can define that action to run five times within one test iteration.

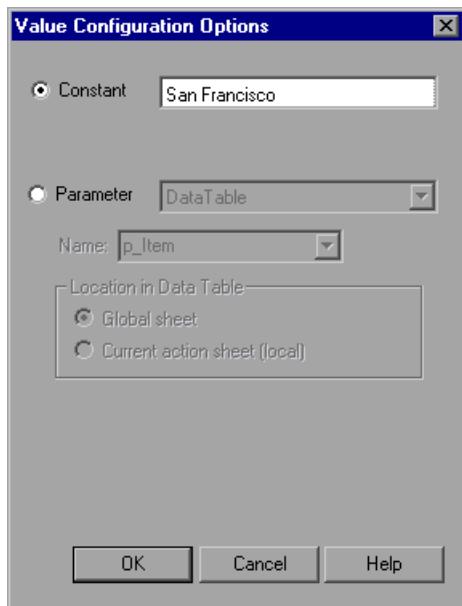
In this section, you will create a data set for the **FlightFinder** action to check how the application handles multiple orders during a single user session.

1 Open the FlightFinder action.

In the Test Flow pane, double-click the **FlightFinder** action. The FlightFinder action opens.

2 Select a step to parameterize.

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.



3 Enter the parameter information.

- a** Select **Parameter** and confirm that the **DataTable** option is selected.
- b** In the **Name** box, type **arrival**.
- c** Select **Global sheet** because you want the iteration to run on the test and not on the action.
- d** Click **OK** to close the Value Configuration Options dialog box.

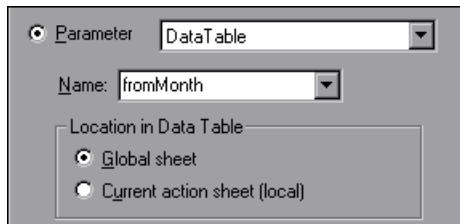
QuickTest automatically adds the **arrival** column to the **Global** data sheet and inserts **San Francisco** into the first cell of the **arrival** column.

4 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.

- a 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. Then, in the **Name** box, type **fromMonth**. Select **Global sheet** in the Value Configuration Options dialog box and click **OK** to close the Value Configuration Options dialog box.



The third column of the Global sheet of the Data Table displays your changes.

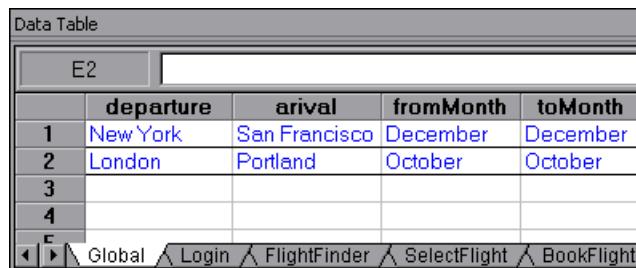
- b In the **toMonth** row in the Keyword View, click the **Value** cell and then click the parameterization icon . The Value Configuration Options dialog box opens.

Select **Parameter** and confirm that the **DataTable** option is selected. In the **Name** box type **toMonth**. Select **Global sheet** in the Value Configuration Options dialog box and click **OK** to close the Value Configuration Options dialog box. The fourth column of the Global sheet of the Data Table displays your changes.

5 Enter the parameterization data in the Data Table.

- a In the Data Table, click the **Global** tab.
- b Type Portland in the second row of the **arrival** column.
- c Type October in the second row of the **fromMonth** and **toMonth** columns.
- d Click an empty cell of the first or second row to save your changes.

The Data Table should appear similar to the following:



A screenshot of a "Data Table" window. The table has four columns: "departure", "arrival", "fromMonth", and "toMonth". Row 1 contains "New York" in "departure", "San Francisco" in "arrival", "December" in "fromMonth", and "December" in "toMonth". Row 2 contains "London" in "departure", "Portland" in "arrival", "October" in "fromMonth", and "October" in "toMonth". Rows 3 and 4 are empty. The "Global" tab is selected at the bottom of the table. Below the table, there are navigation buttons for "Home", "FlightFinder", "SelectFlight", and "BookFlight".

	departure	arrival	fromMonth	toMonth
1	New York	San Francisco	December	December
2	London	Portland	October	October
3				
4				

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 New York 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 Parameterized Test

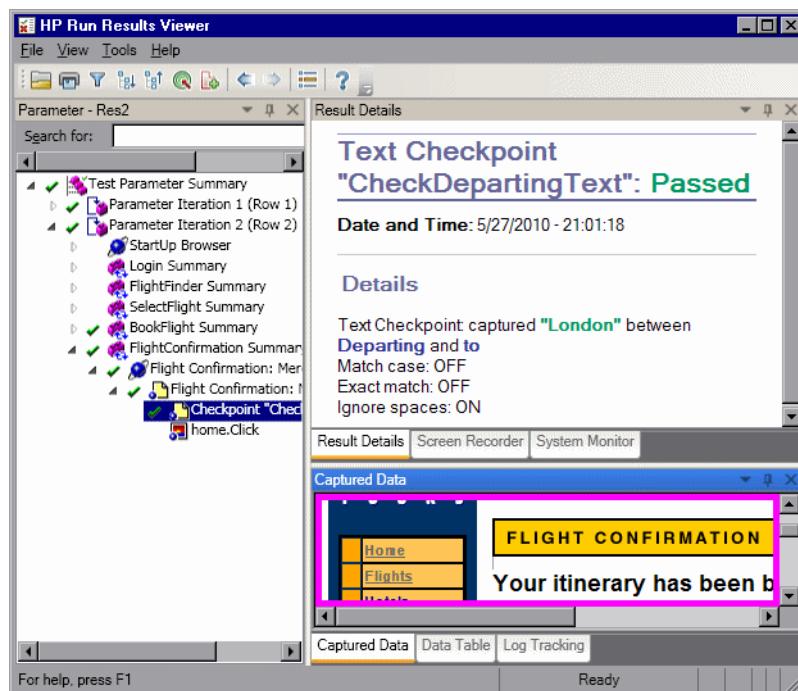
You will now run the modified **Parameter** test. QuickTest will run the test twice, once for each departure city in the Data Table. Each test run is called an **iteration**.

1 Run the Parameter test.

- a Click **Run** on the Automation toolbar or choose **Automation > Run**. The Run dialog box opens.
- b Select **New run results folder** and accept the default results folder name.
- c Click **OK**. When the test run is completed, the Run Results Viewer opens.

2 Examine the results summary.

The Run Results Viewer shows that the iterations of the test passed. Drill down in the run results tree to view the various results.



3 Close the Run Results Viewer.

Choose **File > Exit** to close the Run Results Viewer.

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.

This lesson includes:

- Modifying QuickTest Settings to Save Disk Space on page 129
- Testing Your Own Application—Getting Started on page 130

Modifying QuickTest Settings to Save Disk Space

Some of the settings you configured during this tutorial require a lot of disk space. Therefore, you may want to modify these settings to save disk space.

In Lesson 6, "Running a Test", you configured QuickTest to save all images to the run results. Now, you will configure QuickTest to save images only when a step fails.

1 Open QuickTest.

2 Modify the global settings for saving images to the run results.



- a Choose **Tools > Options** or click the **Options** button, and then click the **Run > Screen Capture** node.
- b In the **Save still image captures to results** option, select **For errors**. (You can also clear the **Save still image captures to results** check box so that no images are ever saved to the run results. However, this may make troubleshooting your tests more difficult.)
- c Click **OK** to close the Options dialog box.

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 Analyze your application.

Determine the development environment to load the relevant QuickTest add-ins and provide support for the objects in your application.

Determine the business processes that users will perform and plan your tests and actions accordingly.

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

At this stage, you can already begin creating the skeletal tests and actions that you will use when testing your application.

2 Prepare your testing infrastructure.

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 with shared object repositories. In this way 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. Although not discussed in this tutorial, you can export test objects from a local object repository to a shared object repository, and you can merge object repositories.

You may also want to create function libraries to enhance QuickTest functionality.

For more information, see the *HP QuickTest Professional User Guide*.

3 Build your test.

While you create your test steps (described in Lesson 5, "Creating a Test"), follow the steps you expect users to perform as they navigate within your application.

4 Enhance your test.

- ▶ Add checkpoints (described in Lesson 7, "Creating Checkpoints and Using Functions") to search for specific values of a page, object, text string, or table cell.
- ▶ Replace fixed values in your test with parameters (described in Lesson 8, "Parameterizing Tests") to check how your application performs the same operations with multiple sets of data.

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

5 Debug your test.

Debug your test to check that it operates smoothly and without interruption. For more information on debugging, see the *HP QuickTest Professional User Guide*.

6 Run your test.

Run your test on your application (described in Lesson 6, "Running and Analyzing Tests") to check that the application functions as expected.

7 Analyze the run 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 run results.)

8 Report defects.

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