Astra® QuickTest™

Tutorial

Version 5.6

MERCURY INTERACTIVE

Astra QuickTest Tutorial, Version 5.6

If you have any comments or suggestions regarding this document, please send them via e-mail to documentation@merc-int.com.

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089 USA

AQTTUT5.6/01

Table of Contents

Welcome to the Astra QuickTest Tutorial

Welcome to the Astra QuickTest tutorial. The tutorial is a self-paced guide that teaches you the basics of testing your Web site with QuickTest, Mercury Interactive's icon-based, functional testing tool for the Web.

QuickTest enables you to test standard Web objects and ActiveX controls. In addition to these environments, QuickTest Professional also enables you to test Java applets and applications and multimedia objects on Web pages.

This tutorial will familiarize you with the process of creating and running automated tests and analyzing the test results.

This introduction describes:

- Using This Tutorial
- Typographical Conventions

Using This Tutorial

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

After completing the tutorial, you can apply the skills you have learned to your own Web site.

Lesson 1, Introducing QuickTest compares automated and manual testing methods. It introduces the QuickTest testing process and familiarizes you with the QuickTest user interface and the sample Mercury Tours Web site.

Lesson 2, Recording Tests teaches you how to record a test and provides an overview of the resulting test that QuickTest displays in the test tree.

Lesson 3, Running and Analyzing Tests shows you how to run a test and view the test results.

Lesson 4, Creating Checkpoints explains how to add checkpoints to your test to check that information in your Web site is displayed as expected.

Lesson 5, Parameterizing Tests shows you how to parameterize your test so that you can run it on multiple sets of data.

Lesson 6, Creating Output Values teaches you how to use output parameters to retrieve data from your Web site.

Lesson 7, Using Regular Expressions teaches you how to create and run a test using regular expressions.

Lesson 8, Dividing Tests into Multiple Actions explains how to divide your test into actions so that you can design more efficient and modular tests.

Lesson 9, Where Do You Go from Here? shows you how to get started testing your own Web site, describes additional features not covered in this tutorial, and tells you where you can find more information about QuickTest.

Typographical Conventions

This book uses the following typographical conventions:

1, 2, 3 Bold numbers indicate steps in a procedure.

➤ Bullets indicate options and features.

> The greater than sign separates menu levels (for example, **File** > **Open**).

Bold **Bold** text indicates object and method names.

Italics *Italic* text indicates variable names.

Example Example font is used for examples and statements that are to be typed literally.

[ ] Square brackets enclose optional parameters.

{ } Curly brackets indicate that one of the enclosed values must be assigned to the current parameter.

... In a line of syntax, an ellipsis indicates that more items of the same format may be included. In a program example, an ellipsis is used to indicate lines of a program that were intentionally omitted.

| A vertical bar indicates that either of the two options separated by the bar should be selected.

1

Introducing QuickTest

In this lesson you will learn about:

- The Benefits of Automated Testing
- The Testing Process
- The QuickTest Window
- The Mercury Tours Sample Web Site

The Benefits of Automated Testing

If you have ever tested 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 Web site goes public. 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 Web site, and then run these tests every time your site changes.

As QuickTest runs tests, it simulates a human user by moving the mouse cursor over a Web page, clicking Graphical User Interface (GUI) objects, and entering keyboard input; but, 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 reacts after repeated execution of the same operations.
Programmable	You can program sophisticated tests that bring out hidden information from the Web site.
Comprehensive	You can build a suite of tests that covers every feature in your Web site.
Reusable	You can reuse tests on different versions of a Web site, even if the user interface changes.

The Testing Process

The QuickTest testing process consists of 6 main phases:

1 **Recording a session on your Web site**

As you navigate through your site, QuickTest graphically displays each *step* you perform in the form of a collapsible icon-based *test tree*. A step is any user action that causes or makes a change in your site, such as clicking a link or image, or entering data in a form.

2 **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 identify whether or not your Web site is functioning correctly.
- Broadening the scope of your test by replacing fixed values with *parameters* lets you check how your Web site performs the same operations with multiple sets of data.
- Adding logic and conditional statements to your test enables you to add sophisticated checks to your test.

3 **Debugging your test**

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

4 **Running your test on a new version of your Web site**

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

5 **Analyzing the test results**

You examine the test results to pinpoint defects in your Web site.

6 **Reporting defects**

If you have TestDirector installed, you can report the defects you discover to a database. TestDirector is Mercury Interactive’s software test management tool.

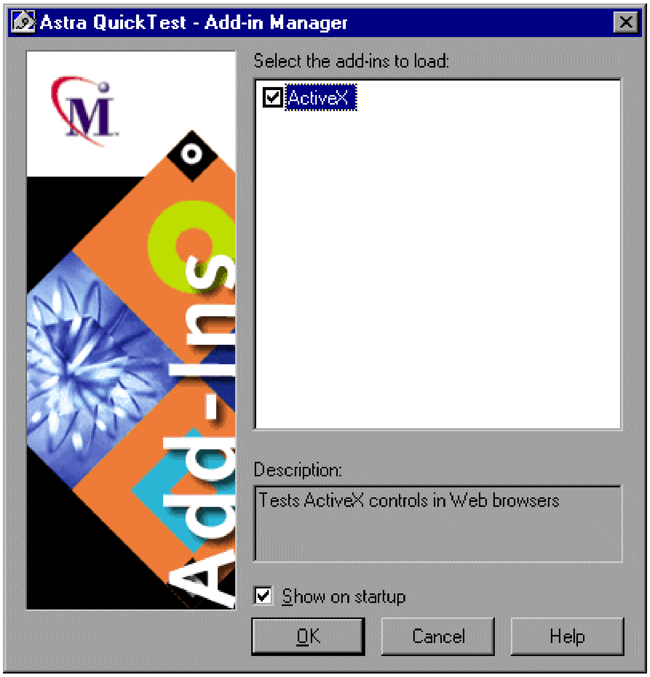
The QuickTest Window

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

To open QuickTest:

Choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**.

When you start QuickTest, the Add-in Manager dialog box opens.



Select the add-ins to load:

☑ ActiveX

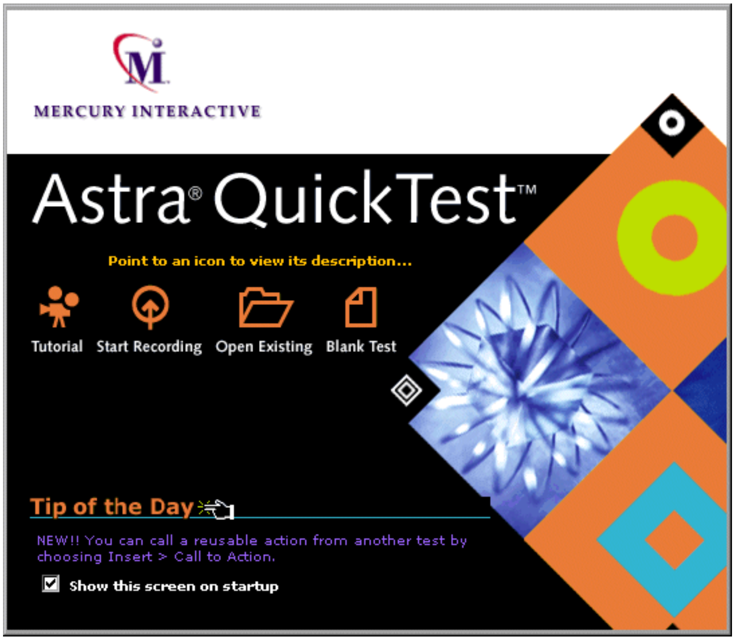
Description:

Tests ActiveX controls in Web browsers

☑ Show on startup

If you do not want this dialog box to open the next time you start QuickTest, clear the **Show on startup** check box.

Click **OK**. The Welcome window opens. From the Welcome window, you can open this tutorial, start recording a new test, open an existing test, or open a new test.



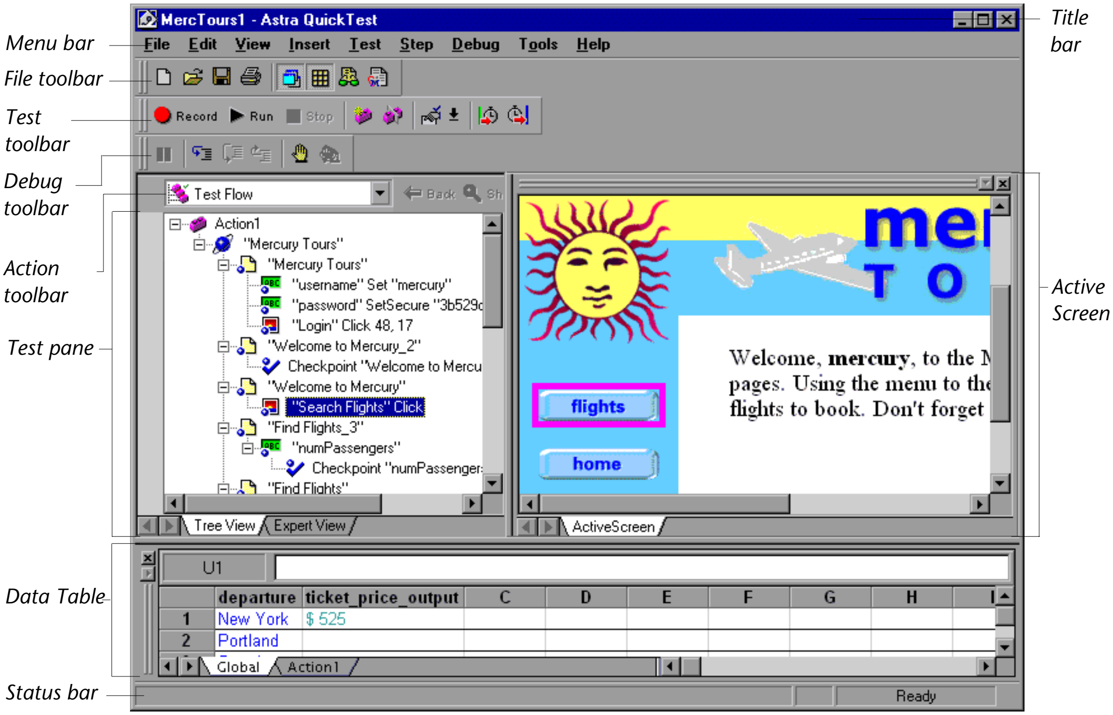
NEW!! You can call a reusable action from another test by choosing Insert > Call to Action.

☑ Show this screen on startup

Tip: You can press the Esc key to close the window and open a blank test.

If you do not want this window to open the next time you start QuickTest, clear the **Show this screen on startup** check box.

When you create a test, the QuickTest window is displayed as follows:



```
Action1
  "Mercury Tours"
    "Mercury Tours"
      "username" Set "mercury"
      "password" SetSecure "3b529c
      "Login" Click 48, 17
    "Welcome to Mercury_2"
      Checkpoint "Welcome to Mercu
    "Welcome to Mercury"
      "Search Flights" Click
    "Find Flights_3"
      "numPassengers"
        Checkpoint "numPassenger:
    "Find Flights"
```

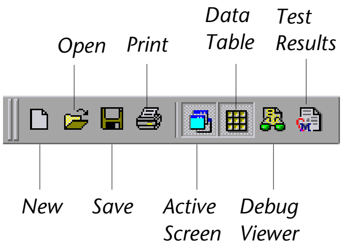
Welcome, mercury, to the M pages. Using the menu to the flights to book. Don't forget

U1

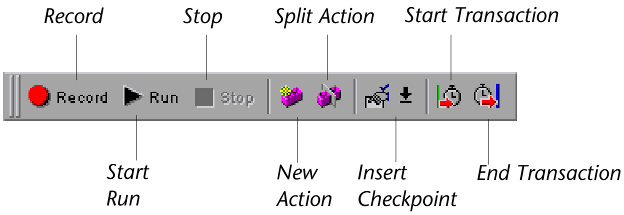
	departure	ticket_price_output	C	D	E	F	G	H
1	New York	$ 525						
2	Portland							

It contains the following key elements:

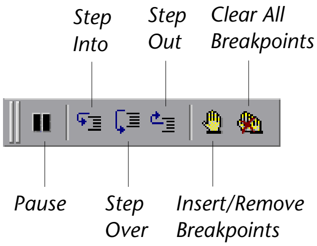
- ➤ *Title bar,* displaying the name of the currently open test
- ➤ *Menu bar,* displaying menus of QuickTest commands
- ➤ *File toolbar,* containing buttons to assist you in managing tests



- *Test toolbar,* containing buttons used while creating and maintaining tests

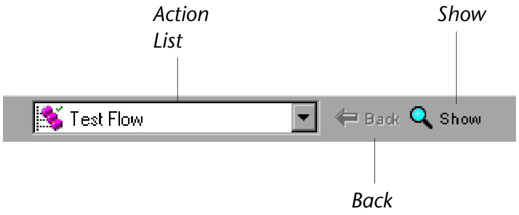


- *Debug toolbar,* containing buttons used while debugging tests



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

- *Action toolbar*, containing 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 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. For additional information, refer to the *Astra QuickTest User's Guide*.

- *Test pane*, containing two tabs to view your test—the Tree View and the Expert View
- *Test Details pane*, containing the Active Screen
- *Data Table*, containing two tabs, Global and Action, to assist you in parameterizing your test
- *Debug Viewer pane*, containing three tabs to assist you in debugging your test—Watch Expressions, Variables, and Command. (The Debug Viewer pane can be opened only when a test run pauses at a breakpoint.)
- *Status bar*, displaying the status of the test

The buttons used in the File and Standard toolbars are described in detail in subsequent lessons. The buttons used in the Debug toolbar are described in detail in the "Debugging Tests" chapter in the *Astra QuickTest User's Guide*. The Action toolbar is described in Lesson 8, "Dividing Tests into Multiple Actions."

Note that you can also execute many menu commands using shortcut keys. For additional information, see the "QuickTest at a Glance" chapter in the *Astra QuickTest User's Guide*.

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

The Mercury Tours Sample Web Site

Mercury Tours is the sample Web application used in the lessons of this tutorial. It simulates a Web-based flight reservation and information service. You should familiarize yourself with this application before starting the tutorial.

To run Mercury Tours:

1 **Launch the Mercury Tours application.**

In your Web browser, type the following URL:
http://mercurytours.mercuryinteractive.com

2 **Sign up with Mercury Tours.**

Log in to Mercury Tours with the **Member name** mercury and the **Password** mercury. Click **login**. The Mercury Tours home page opens.

3 **Explore the Mercury Tours site.**

Click the **flights** button and follow the on-screen instructions to get flight information and reserve a flight. Note that on the Method of Payment page, you do not have to fill out personal or credit card information; simply click **purchase flight** to reserve the flight. You can book several flights.

4 **End your Mercury Tours session.**

When you are done exploring the Mercury Tours application, click **sign off** to log out.

2

Recording Tests

As you navigate through your Web site, QuickTest records your *steps*. These actions are the basis of your test. When you stop recording, you can see the steps of your newly created test in QuickTest in an icon-based tree.

In this lesson you will learn about:

- Recording a Test
- Analyzing the Test Tree

Recording a Test

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

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

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Blank Test**.

If another test is open in QuickTest, choose **File** > **New**. A new test opens.

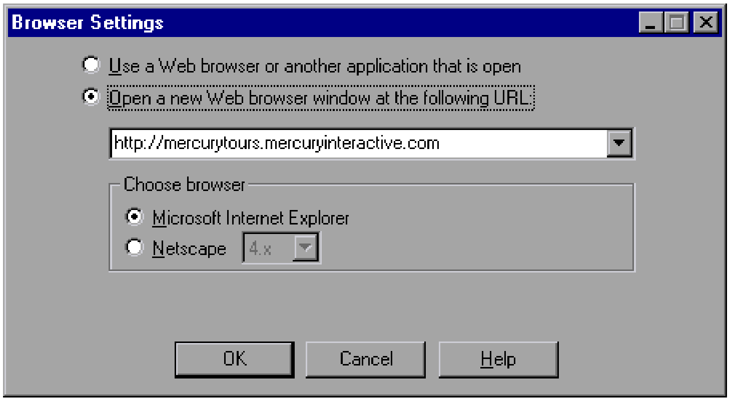
2 Confirm that automatic page checks will not be added.

QuickTest includes a feature that creates automatic page checkpoints. In this lesson, this feature should be disabled.

Choose **Tools** > **Options**. In the Web tab, click the **Advanced** button to open the Advanced Web Options dialog box. Confirm that the **Create a checkpoint for each Web page while recording** check box is cleared. Click **OK**. Click **OK** again to close the Options dialog box.

3 Start recording on the Mercury Tours Web site.

Choose **Test** > **Record** or click the **Record** button. The Browser Settings dialog box opens.



Browser Settings

○ Use a Web browser or another application that is open

● Open a new Web browser window at the following URL:

http://mercurytours.mercuryinteractive.com

Choose browser

● Microsoft Internet Explorer

○ Netscape 4.x

Accept the URL (as it is displayed above) and choose a browser. Click **OK**. Your browser opens and displays the Mercury Tours Web site.

4 Login to the Mercury Tours Web site.

In the **Member name** edit field, type mercury. In the **Password** edit field, type mercury. Click **login**. The Welcome page opens.

5 Book a flight.

- Click **flights**. The Find Flight page opens.
- In the **Departure City** list, select **New York**.
- In the **Arrival City** list, select **San Francisco**.
- Under **Type of Seat**, click **First**.

- ➤ Click **continue...**. The Search Results Flights page opens.
- ➤ Click **continue...** to accept the default flight. The Method of Payment page opens.
- ➤ Click **purchase flight**. The Flight Confirmation page opens.
- ➤ Click **book another flight**. The Find Flight page opens.

6 Stop recording.

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

7 Save your test.

Select **File** > **Save** or click the **Save** button. The Save QuickTest Test dialog box opens.

Type Lesson2 in the **File name** field and click **Save**. QuickTest displays the test name in the title bar.

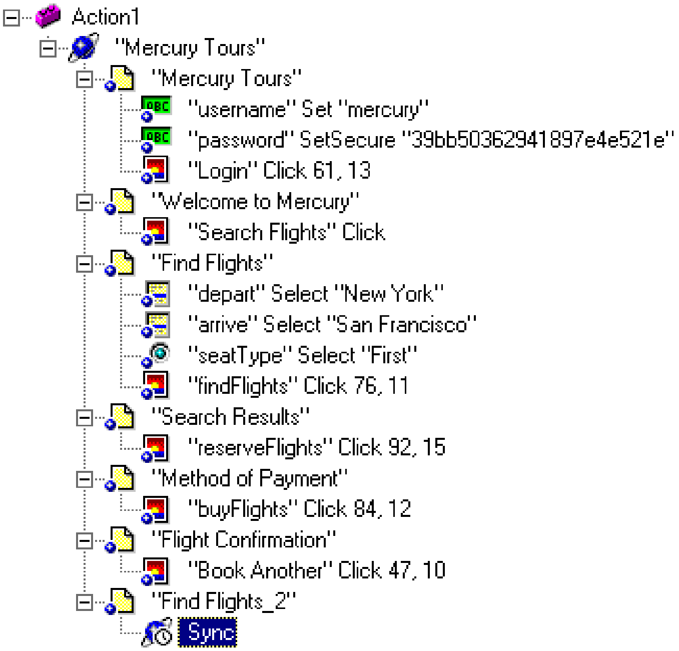
You have now reserved a first class ticket from New York to San Francisco. QuickTest recorded your steps from the moment you clicked the Record button until the moment you clicked the Stop button.

Analyzing the Test Tree

QuickTest displays the test you recorded in the *test tree* in the Tree View. As you record your test, QuickTest generates steps in the test tree. Each step you performed while recording is represented in the test tree.

Your test tree displays seven Web pages. You can expand (+) or collapse (-) the steps under each Web page.

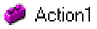
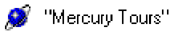
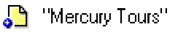
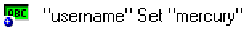
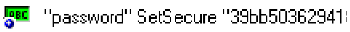
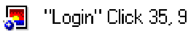
You can expand the entire test tree by choosing **View** > **Expand All**. When you expand your test tree, it displays similar to the following:



```
Action1
  "Mercury Tours"
    "Mercury Tours"
      "username" Set "mercury"
      "password" SetSecure "39bb50362941897e4e521e"
      "Login" Click 61, 13
    "Welcome to Mercury"
      "Search Flights" Click
    "Find Flights"
      "depart" Select "New York"
      "arrive" Select "San Francisco"
      "seatType" Select "First"
      "findFlights" Click 76, 11
    "Search Results"
      "reserveFlights" Click 92, 15
    "Method of Payment"
      "buyFlights" Click 84, 12
    "Flight Confirmation"
      "Book Another" Click 47, 10
    "Find Flights_2"
      Sync
```

In this section you will explore the steps performed on the first page of the test. When you perform a step on an object, QuickTest displays the step in icon form in the test tree. Each step is represented by an icon and a title.

The table below provides an explanation of the steps in a page in the test tree.

Step	**Description**
 Action1	*Action1* is the **action** name. Actions are described in Lesson 8, “Dividing Tests into Multiple Actions.”
 "Mercury Tours"	The **browser** loads the *Mercury Tours* Web site.
 "Mercury Tours"	The name of the **Web page**.
 "username" Set "mercury"	*username* is the name of the **edit field**. *Set* is the method performed on the edit field. *mercury* is the value of the edit field.
 "password" SetSecure "39bb50362941	*password* is the name of the **edit field**. *SetSecure* is an encrypt method performed on the edit field. *39bb50362941897e4e521e* is the encrypted value of the password.
 "Login" Click 35, 9	*Login* is the name of the **image**. *Click* is the method performed on the image. *35, 9* are the x- and y-coordinates where the image was clicked.

You can examine all the steps in this way. Take a few minutes to examine the test tree in further detail.

3

Running and Analyzing Tests

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

In this lesson you will learn about:

➤ Running a Test

➤ Analyzing Test Results

Running a Test

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

1 Start QuickTest and open the *Lesson2* test.

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

➤ If the Welcome window opens, click **Open Existing**, locate the *Lesson2* test, and click **Open** in the **Open QuickTest Test** dialog box.

➤ Otherwise choose **File** > **Open** or click the **Open** button, locate the *Lesson2* test, and click **Open** in the **Open QuickTest Test** dialog box.

2 Confirm that all images will be saved to the test results.

QuickTest includes a feature that determines when to save images to the test results. In this lesson, images should be saved to the test results always.

Choose **Tools** > **Options**. Click the **Run** tab. Select **always** in the **Save step screen capture to test results** list box.

3 Confirm that the execution arrow will be displayed during the test run.

When running a test, QuickTest displays a yellow arrow in the left margin of the test tree, indicating the step that is running.

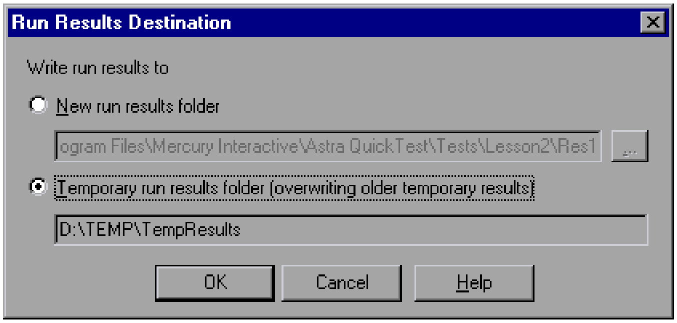
From the **Run** tab of the Options dialog box, select the **Display execution arrow (Slow mode)** option. Click **OK** to close the Options dialog box.

4 Start running your test.

▶ Run

Click **Run** or choose **Test** > **Run**. The Run Results Destination dialog box opens.

The Run Results Destination dialog box enables you to configure the file location settings for your results. Select **Temporary run results folder**, and click **OK**.



Run Results Destination

Write run results to

○ New run results folder

ogram Files\Mercury Interactive\Astra QuickTest\Tests\Lesson2\Res1

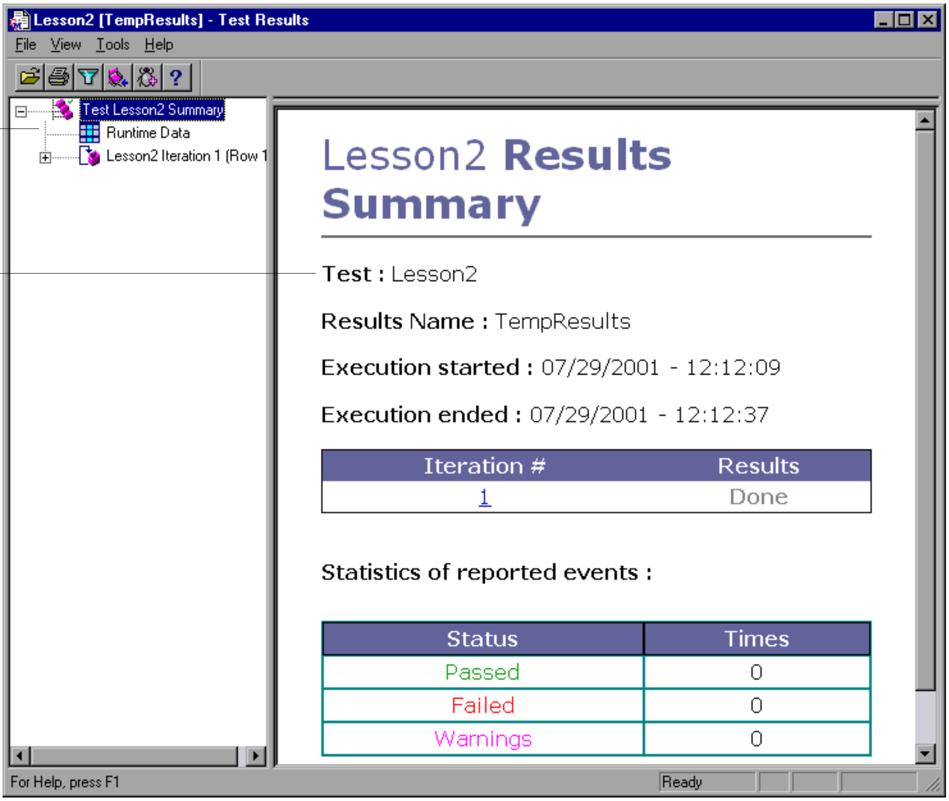
● Temporary run results folder (overwriting older temporary results)

D:\TEMP\TempResults

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

Analyzing Test Results

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



Test results tree

Results Summary pane

Lesson2 Results Summary

Test : Lesson2

Results Name : TempResults

Execution started : 07/29/2001 - 12:12:09

Execution ended : 07/29/2001 - 12:12:37

Iteration #	Results
1	Done

Statistics of reported events :

Status	Times
Passed	0
Failed	0
Warnings	0

Initially, the Test Results window contains 2 panes, the Tree pane and the Results Summary pane:

- The **Tree pane** contains the test results tree, which is an icon-based view of the steps that were performed while the test was running. The test results tree, similar to the test tree, is organized according to the Web pages visited during the iteration. The steps performed during the test run are represented by icons in the tree. Each iteration is numbered. (The test you ran had only one iteration.)

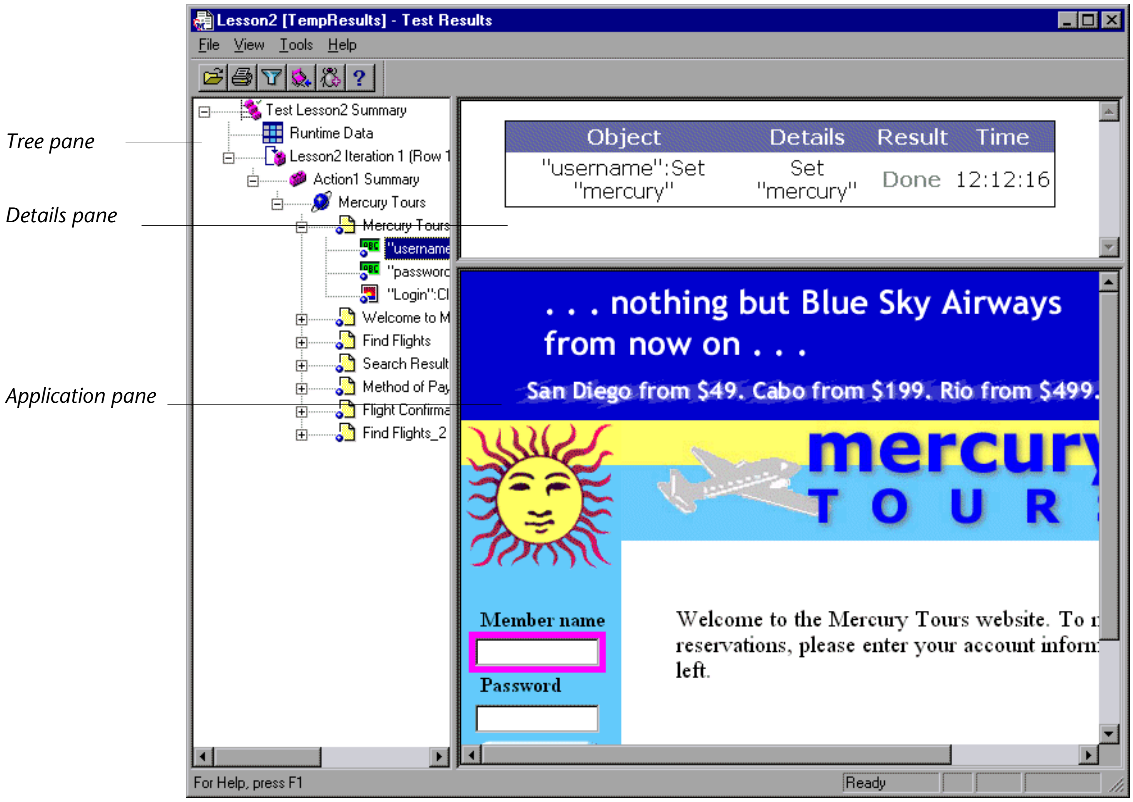
➤ The **Results Summary pane** contains the details of the test. The Execution table indicates which iterations passed and which failed. The statistics table indicates the number of checkpoints or reports that passed, failed, or received warnings during the test.

The test passed because QuickTest was able to navigate the Mercury Tours site according to the test that was originally recorded.

1 View the test results for a specific step.

In the test results tree, expand **(+) Test Lesson2 Summary** > **Lesson2 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Mercury Tours** > **"username":Set "mercury"**.

Highlight **"username":Set "mercury"** in the test results tree.



Object	Details	Result	Time
"username":Set "mercury"	Set "mercury"	Done	12:12:16

. . . nothing but Blue Sky Airways from now on . . .

San Diego from $49. Cabo from $199. Rio from $499.

Welcome to the Mercury Tours website. To r reservations, please enter your account inform left.

The Test Results window now contains three panes:

- The **Tree pane** contains the test results tree.
- The **Details pane** displays the details of the step performed during the test run.
- The **Application pane** displays a screen capture of the Web page on which the step was performed. When you click a page in the test results tree, QuickTest displays the corresponding page in the Application pane. When you click a step in the test results tree, the corresponding object is highlighted in the Application pane.

2 Close the Test Results window.

Choose **File** > **Exit**.

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

4

Creating Checkpoints

In the previous lessons, you created and ran a test that checks that a series of steps 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 learn about:

- Understanding Checkpoint Types
- Checking Pages
- Checking Web Objects
- Checking Text
- Checking Tables
- Running and Analyzing a Test with Checkpoints

Understanding Checkpoint Types

Astra QuickTest offers seven types of checkpoints for standard Web objects:

- A *page checkpoint* checks the characteristics of a Web page.

  For example: You can check how long a Web page takes to load or whether a Web page contains broken links.

- An *accessibility checkpoint* identifies areas of your Web site that require special attention according to the W3C Web Content Accessibility Guidelines, in order to help you make your application Section 508 compliant.

For example: You can check whether the images on your Web page include the ALT tags required by the W3C Web Content Accessibility Guidelines.

- A *text checkpoint* checks that a text string is displayed in the appropriate place on a Web page.

  For example: Suppose the Method of Payment page displays the sentence Please enter the passenger and payment details for flight 360. You can create a text checkpoint that checks that the words "flight 360" are displayed in the appropriate place.

- An *object checkpoint* checks the values of an object property on a Web page.

  For example: In the Find Flight page, suppose you select the First Class radio button. You can add an object checkpoint that checks that the radio button is activated.

- An *image checkpoint* checks the values of an image property on a Web page.

  For example: Suppose you select the animated banner "Mercury Tours" at the top of any page of the Web site. You can add an image checkpoint that checks that the image source file is correct. Note that this tutorial does not describe how to create an image checkpoint. For additional information, refer to the "Checking Web Objects" chapter in the *Astra QuickTest User's Guide*.

- A *table checkpoint* checks information within a table on a Web page.

  For example: In the Search Results Flights page, a table lists all the available flights for a specific route. Suppose you are booking a flight from New York to San Francisco. You can add a table checkpoint that checks that the time of the first flight from New York to San Francisco is correct.

- A *database checkpoint* checks the contents of databases accessed by your Web site.

  For example: You can use a database checkpoint to check the contents of the database containing flight information for the Mercury Tours Web site. Note that this tutorial does not describe how to create a database checkpoint. For additional information, refer to the "Checking Databases" chapter in the *Astra QuickTest User's Guide*.

Note that you can add checkpoints to your test while recording or afterward. This lesson explains how to create checkpoints after recording a test. For information on how to create checkpoints while recording, refer to the "Creating Checkpoints" chapter in the *Astra QuickTest User's Guide*.

Checking Pages

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

1 **Start QuickTest and open the *Lesson2* test.**

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Open Existing**.

Browse to the *Lesson2* test. Click **Open**.

2 **Save the test as *Lesson4*.**

Select **File** > **Save As**. Save the test as Lesson4.

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

If you do not see the Active Screen on the right side of the screen, toggle the Active Screen toolbar button or select the Active Screen option from the View menu to enable it. Make sure the Active Screen tab is selected.

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

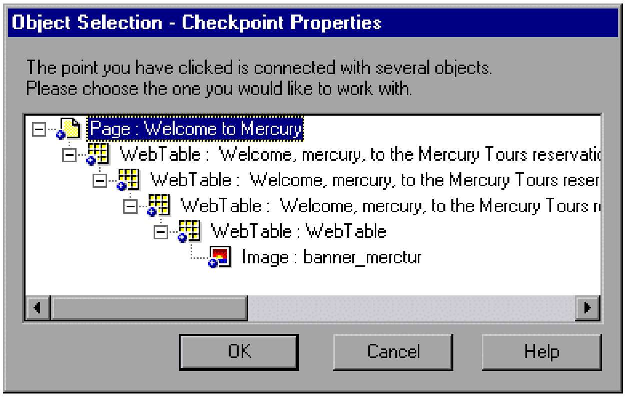
In the test tree, expand **(+) Action1** > **"Mercury Tours"** > **"Welcome to Mercury"**.

Highlight the **Welcome to Mercury** page in the test tree. The Welcome page is displayed in the Active Screen.

5 Create the page checkpoint.

Right-click anywhere in the Active Screen and choose **Insert Checkpoint**. The Object Selection - Checkpoint Properties dialog box opens. (Note that this dialog box may include some different elements, depending where you click in the Active Screen.)

Highlight **Page: Welcome to Mercury** (the top element in the tree).

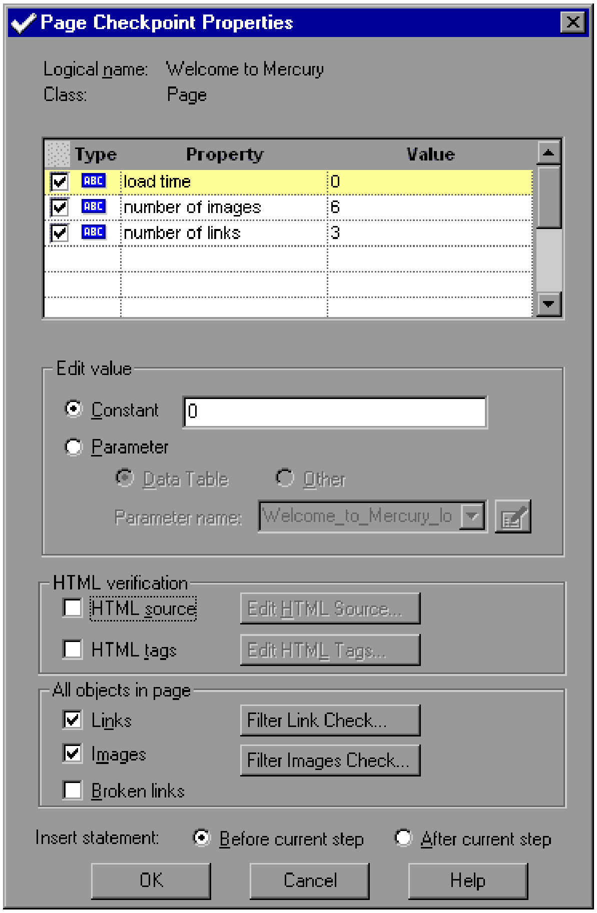


Object Selection - Checkpoint Properties

The point you have clicked is connected with several objects.
Please choose the one you would like to work with.

- Page : Welcome to Mercury
  - WebTable : Welcome, mercury, to the Mercury Tours reservatic
    - WebTable : Welcome, mercury, to the Mercury Tours reser
      - WebTable : Welcome, mercury, to the Mercury Tours r
        - WebTable : WebTable
          - Image : banner_merctur

Click **OK**. The Page Checkpoint Properties dialog box opens.



Page Checkpoint Properties

Logical name: Welcome to Mercury
Class: Page

Type	Property	Value
☑ ABC	load time	0
☑ ABC	number of images	6
☑ ABC	number of links	3

Edit value
● Constant 0
○ Parameter
● Data Table ○ Other
Parameter name: Welcome_to_Mercury_lo

HTML verification
□ HTML source
□ HTML tags

All objects in page
☑ Links
☑ Images
□ Broken links

Insert statement: ● Before current step ○ After current step

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

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

Note: You can click the Filter Link Check button to open the Filter Link Checks dialog box, where the names and URLs for each link are displayed. QuickTest checks each link that is selected. You can click the Filter Images Check button to open the Filter Image Check dialog box, where the names and sources for each image are displayed. QuickTest checks each image that is selected. For additional information about these dialog boxes, refer to the "Checking Web Objects" chapter in the *Astra QuickTest User's Guide*.

Click **OK**.

QuickTest adds the page checkpoint to your test. It is displayed in the test tree as a new step under the Welcome to Mercury page.

6 **Save the test.**

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

Checking Web Objects

In this part of the lesson, you will add an object checkpoint that checks the value in the No. of Passengers edit field in the Find Flights page.

1 **Locate the page where you want to add an object checkpoint.**

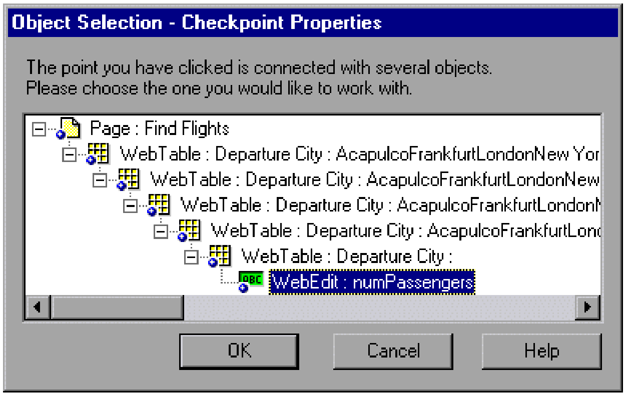
In the test tree, expand **(+) Action1** > **"Mercury Tours"** > **"Find Flights"**.

Highlight the **Find Flights** page in the test tree. The Find Flight page is displayed in the Active Screen.

2 **Create the object checkpoint.**

In the Active Screen, right-click the **No. of Passengers** box and choose **Insert Checkpoint**. The Object Selection - Checkpoint Properties dialog box opens.

Confirm that **WebEdit: numPassengers** is highlighted.

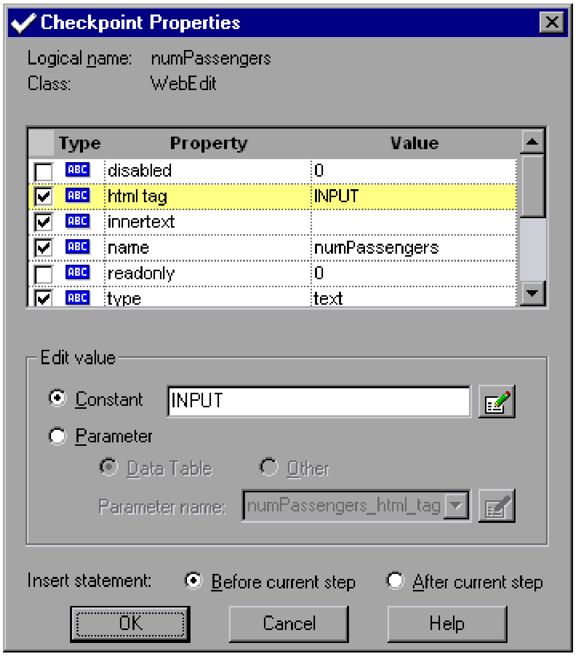


Object Selection - Checkpoint Properties

The point you have clicked is connected with several objects.
Please choose the one you would like to work with.

- Page : Find Flights
  - WebTable : Departure City : AcapulcoFrankfurtLondonNew Yor
    - WebTable : Departure City : AcapulcoFrankfurtLondonNew
      - WebTable : Departure City : AcapulcoFrankfurtLondonN
        - WebTable : Departure City : AcapulcoFrankfurtLond
          - WebTable : Departure City :
            - WebEdit : numPassengers

Click **OK**. The Checkpoint Properties dialog box opens.



Checkpoint Properties

Logical name: numPassengers
Class: WebEdit

	Type	Property	Value
□	ABC	disabled	0
☑	ABC	html tag	INPUT
☑	ABC	innertext	
☑	ABC	name	numPassengers
□	ABC	readonly	0
☑	ABC	type	text

Edit value

● Constant INPUT

○ Parameter

● Data Table ○ Other

Parameter name: numPassengers_html_tag

Insert statement: ● Before current step ○ After current step

This dialog box displays the properties of the object:

- The **Logical 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 field.
- The **ABC** icon indicates that the value of the property is a constant.

The table below describes the selected, default checks.

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	numPassengers	"numPassengers" is the name of the edit field.
type	text	"text" is the type of object as defined in the HTML source code.
value	1	"1" is the value 1 in the edit field.

For each object class, QuickTest recommends default property checks. Click **OK** to accept the default properties for the object (html tag, innertext, name, type, and value).

QuickTest adds the object checkpoint to your test. It is displayed in the test tree as a new step under the Find Flights page.

3 Save the test.

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

Checking Text

In this part of the lesson, you will add a text checkpoint to your test that checks whether "New York" is displayed in the Search Results Flights page.

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

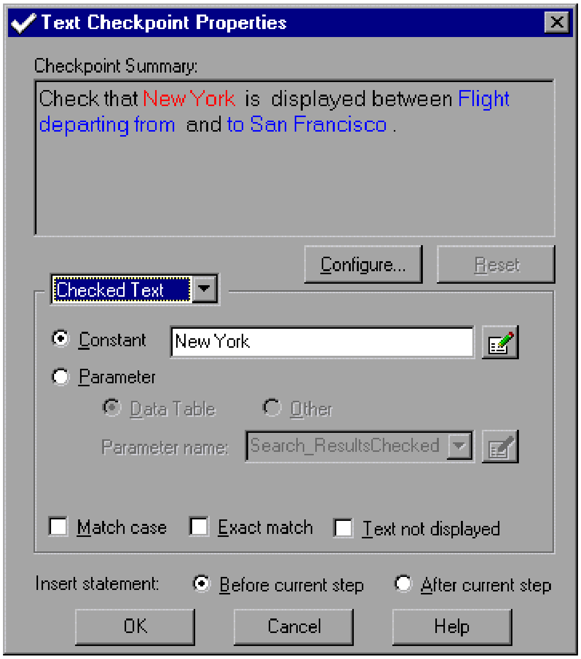
In the test tree, expand **(+) Action1** > **"Mercury Tours"** > **"Search Results"**.

Highlight the **Search Results** page in the test tree. The Search Results Flights page is displayed in the Active Screen.

2 Create the text checkpoint.

In the Active Screen, highlight the text **New York**.

Right-click the highlighted text and choose **Insert Text Checkpoint.** The Text Checkpoint Properties dialog box opens.



Text Checkpoint Properties

Checkpoint Summary:

Check that New York is displayed between Flight departing from and to San Francisco .

Checked Text

● Constant New York

○ Parameter

● Data Table ○ Other

Parameter name: Search_ResultsChecked

□ Match case □ Exact match □ Text not displayed

Insert statement: ● Before current step ○ After current step

In the **Checked Text** box, the **Constant** field displays the highlighted text string. This is the text QuickTest looks for when running the test.

The list box contains options for setting the text before and the text after. Accept the settings for Text Before and Text After.

Click **OK** to accept the default settings in the Checked Text box.

QuickTest adds the text checkpoint to your test. It is displayed in the test tree as a new step under the Search Results page.

3 **Save the test.**

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

Checking Tables

In this part of the lesson, you will add a table checkpoint that checks the cost of the flight in the Flight Confirmation page.

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

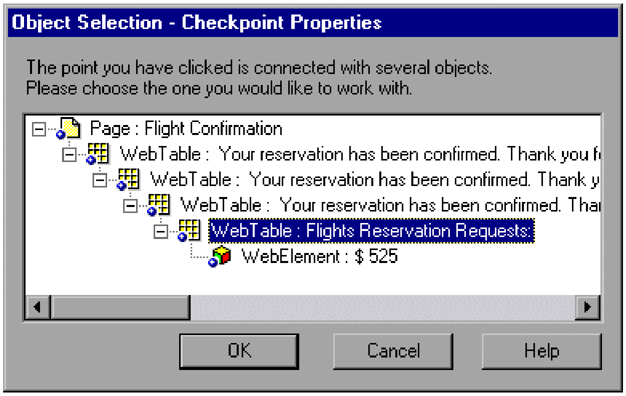
In the test tree, expand **(+) Action1** > **“Mercury Tours”** > **“Flight Confirmation”**.

Highlight the **Flight Confirmation** page in the test tree. The Flight Confirmation page is displayed in the Active Screen.

2 **Create the table checkpoint.**

In the Active Screen, right-click the Total Cost of the ticket ($525) in the Cost column (in the grey **Flight Reservation Requests** table) and choose **Insert Checkpoint**. The Object Selection - Checkpoint Properties dialog box opens.

Make sure that **WebTable: Flight Reservation Requests** is highlighted.

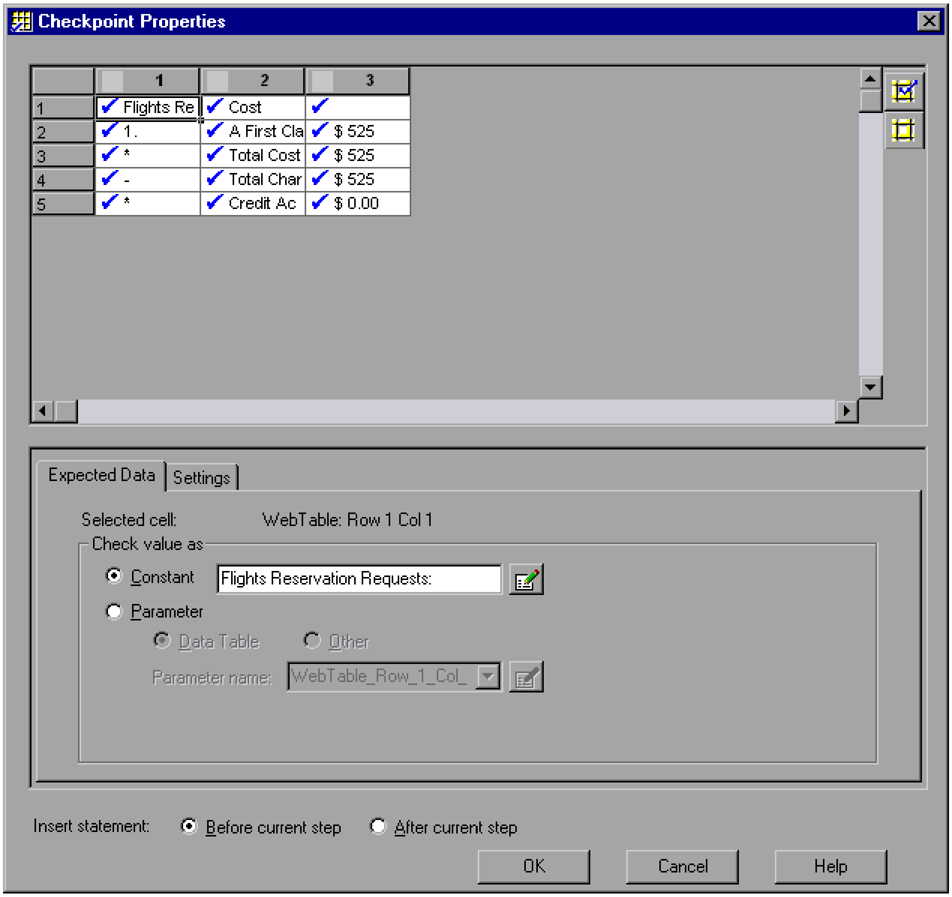


Object Selection - Checkpoint Properties

The point you have clicked is connected with several objects.
Please choose the one you would like to work with.

- Page : Flight Confirmation
  - WebTable : Your reservation has been confirmed. Thank you f
    - WebTable : Your reservation has been confirmed. Thank y
      - WebTable : Your reservation has been confirmed. Thai
        - WebTable : Flights Reservation Requests:
          - WebElement : $ 525

Click **OK**. The Checkpoint Properties dialog box opens. The dialog box displays the rows and columns of the table.



Checkpoint Properties

	1	2	3
1	✓ Flights Re	✓ Cost	✓
2	✓ 1.	✓ A First Cla	✓ $ 525
3	✓ *	✓ Total Cost	✓ $ 525
4	✓ -	✓ Total Char	✓ $ 525
5	✓ *	✓ Credit Ac	✓ $ 0.00

Selected cell: WebTable: Row 1 Col 1

Check value as

● Constant: Flights Reservation Requests:

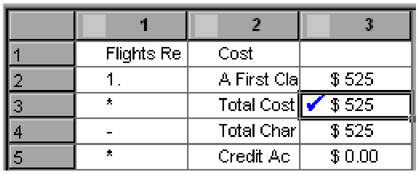
○ Parameter

● Data Table ○ Other

Parameter name: WebTable_Row_1_Col_

Insert statement: ● Before current step ○ After current step

By default, QuickTest adds a check mark to all cells. Clear the check marks by double-clicking the column headers. Add a check mark to cell 3 in row 3 by double-clicking the cell. QuickTest checks only the cells containing checkmarks.



	1	2	3
1	Flights Re	Cost	
2	1.	A First Cla	$ 525
3	*	Total Cost	✓ $ 525
4	-	Total Char	$ 525
5	*	Credit Ac	$ 0.00

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

Accept the settings in the Settings tab and click **OK** to close the dialog box.

QuickTest adds the table checkpoint to your test. It is displayed in the test tree as a new step under the Flight Confirmation page.

3 Save the test.

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

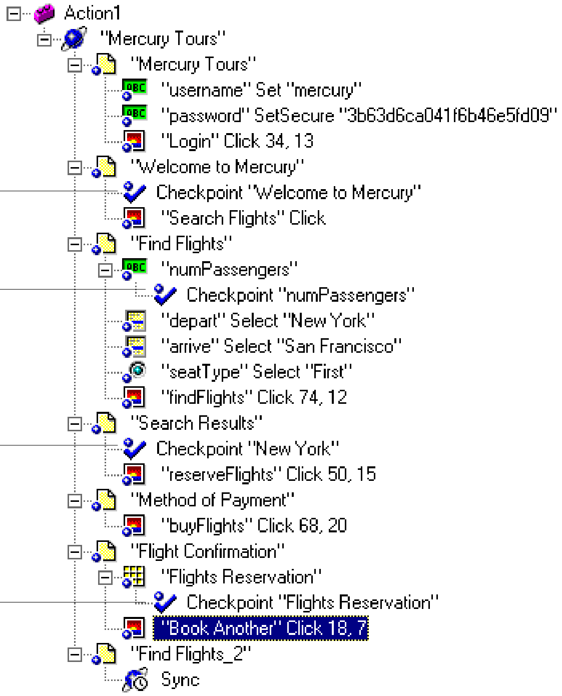
Running and Analyzing a Test with Checkpoints

In this part of the lesson you will review the test to which you added checkpoints, run the test, and analyze the checkpoint results.

1 **Expand the test tree and review your test.**

Choose **View** > **Expand All**.

The test tree is similar to the following:



```
Action1
  "Mercury Tours"
    "Mercury Tours"
      "username" Set "mercury"
      "password" SetSecure "3b63d6ca041f6b46e5fd09"
      "Login" Click 34, 13
    "Welcome to Mercury"
      Checkpoint "Welcome to Mercury"
      "Search Flights" Click
    "Find Flights"
      "numPassengers"
        Checkpoint "numPassengers"
      "depart" Select "New York"
      "arrive" Select "San Francisco"
      "seatType" Select "First"
      "findFlights" Click 74, 12
    "Search Results"
      Checkpoint "New York"
      "reserveFlights" Click 50, 15
    "Method of Payment"
      "buyFlights" Click 68, 20
    "Flight Confirmation"
      "Flights Reservation"
        Checkpoint "Flights Reservation"
      "Book Another" Click 18, 7
    "Find Flights_2"
      Sync
```

Page checkpoint

Object checkpoint

Text checkpoint

Table checkpoint

2 **Start running your test.**

Click **Run** or choose **Test** > **Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. QuickTest opens your browser and begins to run the test.

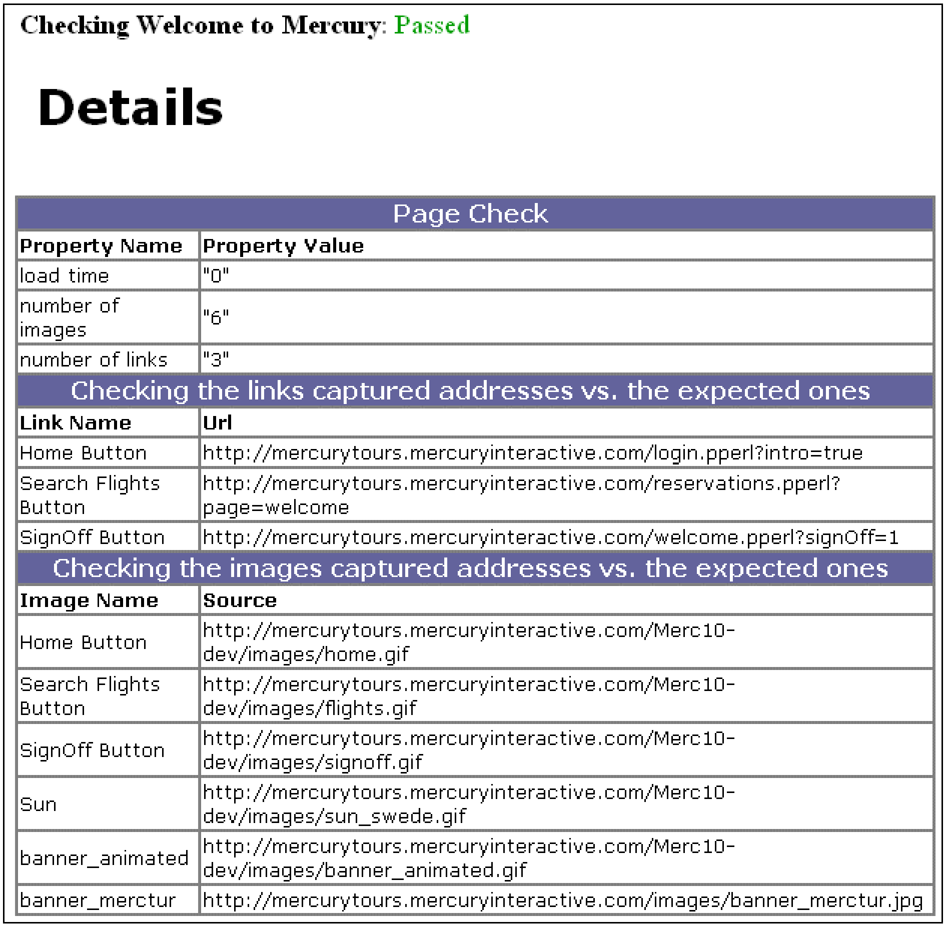
3 **View the test results.**

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

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

In the test results tree, expand (+) **Test Lesson4 Summary** > **Lesson4 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Welcome to Mercury**. Highlight **Checkpoint:“Welcome to Mercury”**.

In the Details pane (the top right pane of the Test Results window), you can review the details of the page checkpoint. It lists the checks that were performed in the page checkpoint. The checkpoint passed because the actual values of the checked properties are the same as the expected values.



Checking Welcome to Mercury: Passed

Details

Page Check	
Property Name	**Property Value**
load time	"0"
number of images	"6"
number of links	"3"

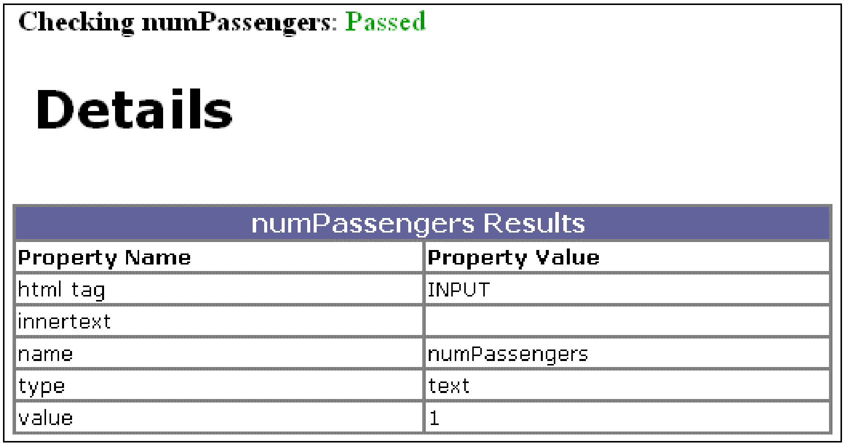
Checking the links captured addresses vs. the expected ones	
Link Name	**Url**
Home Button	http://mercurytours.mercuryinteractive.com/login.pperl?intro=true
Search Flights Button	http://mercurytours.mercuryinteractive.com/reservations.pperl?page=welcome
SignOff Button	http://mercurytours.mercuryinteractive.com/welcome.pperl?signOff=1

Checking the images captured addresses vs. the expected ones	
Image Name	**Source**
Home Button	http://mercurytours.mercuryinteractive.com/Merc10-dev/images/home.gif
Search Flights Button	http://mercurytours.mercuryinteractive.com/Merc10-dev/images/flights.gif
SignOff Button	http://mercurytours.mercuryinteractive.com/Merc10-dev/images/signoff.gif
Sun	http://mercurytours.mercuryinteractive.com/Merc10-dev/images/sun_swede.gif
banner_animated	http://mercurytours.mercuryinteractive.com/Merc10-dev/images/banner_animated.gif
banner_merctur	http://mercurytours.mercuryinteractive.com/images/banner_merctur.jpg

5 View the results of the object checkpoint.

In the test results tree, expand (+) **Test Lesson4 Summary** > **Lesson4 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Find Flights** > **numPassengers**. Highlight **Checkpoint:“numPassengers”**.

In the Details pane (the top right pane of the Test Results window), you can review the details of the object checkpoint. It lists the properties that were checked and the values of the properties. The checkpoint passed because the actual values of the checked properties are the same as the expected values.



Checking numPassengers: Passed

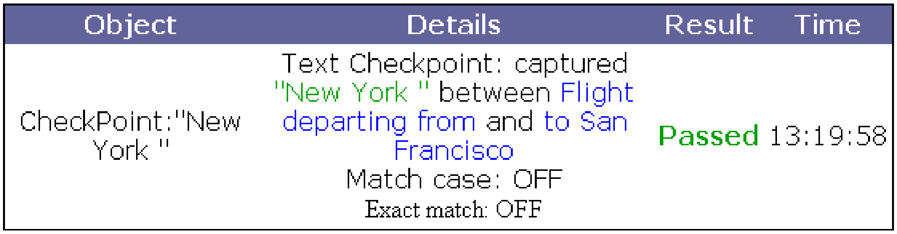
Details

numPassengers Results	
Property Name	**Property Value**
html tag	INPUT
innertext	
name	numPassengers
type	text
value	1

6 View the results of the text checkpoint.

In the test results tree, expand (+) **Test Lesson4 Summary** > **Lesson4 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Search Results**. Highlight **Checkpoint:"New York"**.

In the Details pane (the top right pane of the Test Results window), you can review the details of the text checkpoint. The checkpoint passed because the actual text is the same as the expected text.

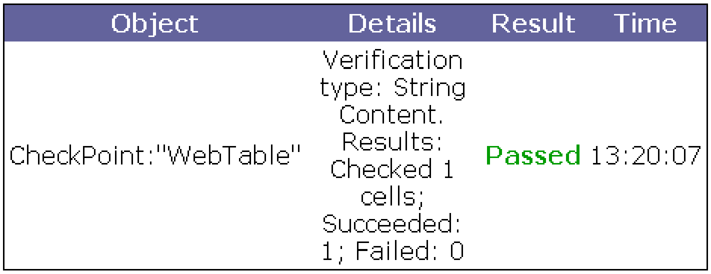


Object	Details	Result	Time
CheckPoint:"New York "	Text Checkpoint: captured "New York " between Flight departing from and to San Francisco Match case: OFF Exact match: OFF	Passed	13:19:58

7 View the results of the table checkpoint.

In the test results tree, expand (+) **Test Lesson4 Summary** > **Lesson4 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Flight Confirmation** > **Flights Reservation**. Highlight **Checkpoint:"Flights Reservation"**.

In the Details pane (the top right pane of the Test Results window), you can review the details of the table checkpoint. The checkpoint passed because the actual value of the checked cell is the same as the expected value.



Object	Details	Result	Time
CheckPoint:"WebTable"	Verification type: String Content. Results: Checked 1 cells; Succeeded: 1; Failed: 0	Passed	13:20:07

8 Close the Test Results window.

Choose **File** > **Exit**.

5

Parameterizing Tests

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

Note: You can also parameterize your test with other types of parameters in order to use random numbers or environment variable values. For more information, refer to the *Astra QuickTest User's Guide*.

In this lesson you will learn about:

- ➤ Defining a Data Table Parameter
- ➤ Assigning Parameter Values into a Data Table
- ➤ Running and Analyzing a Parameterized Test
- ➤ Modifying a Parameterized Test

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 *Lesson4* test.**

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Open Existing**.

Browse to the *Lesson4* test. Click **Open**.

2 **Save the test as *Lesson5*.**

Select **File** > **Save As**. Save the test as Lesson5.

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

If you do not see the Active Screen on the right side of the screen, toggle the Active Screen toolbar button or select the Active Screen option from the View menu to enable it. Make sure the Active Screen tab is selected.

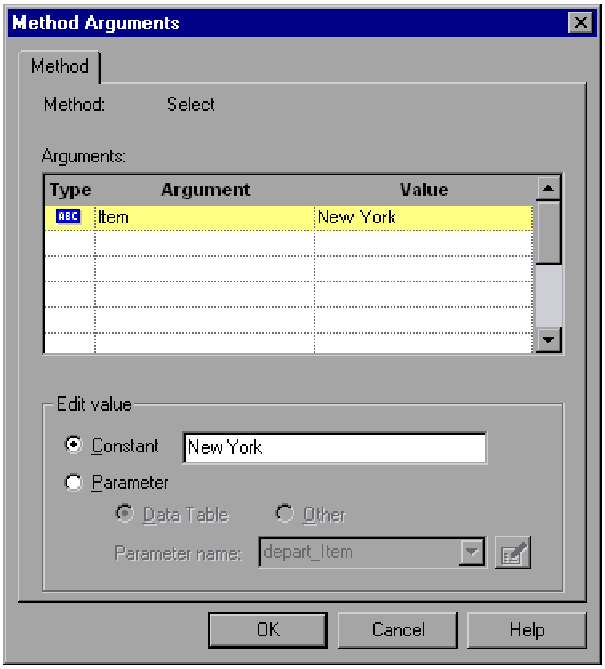
4 **Confirm that the Data Table option is enabled.**

If you do not see the Data Table at the bottom of the screen, toggle the **Data Table** toolbar button or select the Data Table option from the View menu to enable it.

5 Select the text to parameterize.

In the test tree, expand (+) **Mercury Tours** > **Find Flights** > "**depart**" **Select** "**New York**".

Right-click "**depart**" **Select** "**New York**" and select **Method Arguments**. The Method Arguments dialog box opens.



Method Arguments

Method: Select

Arguments:

Type	Argument	Value
ABC	Item	New York

Edit value

● Constant: New York

○ Parameter

● Data Table ○ Other

Parameter name: depart_Item

6 Set the parameterization properties.

A *method* is the term used in VBScript for an action performed on an object. In this case the method is "select" because an item was selected from a list. The Arguments table displays the argument's properties.

- The **ABC** icon indicates that the argument is a constant.
- **Item** indicates that the argument is an item from a list.
- **New York** is the value of the argument.

Highlight the **Item** row if it is not already highlighted. The Edit Value box is enabled and displays the constant value New York.

In the Edit value box, select **Parameter** to use a parameter in place of "New York". Confirm that the **Data Table** parameter option is selected. The Parameter name box is enabled and displays the default name "depart_Item".

In the Parameter name box, replace depart_Item with departure. Click **OK** to close the dialog box. QuickTest automatically adds the departure parameter to the Data Table and inserts New York into the first row of the departure column.

Note the difference in the step. Previously the step was displayed as "depart" Select "New York". Now the step is displayed as "depart" Select <departure> and the <#> icon is displayed beside the step. This indicates that the step is parameterized and the value to be used for the select method should be taken from the departure column of the Data Table.

Assigning Parameter Values into a Data Table

QuickTest displays parameter values in the Data Table. In this section you will add two more cities to the departure column.

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

Click row 2 in the departure column and type Portland.

Click row 3 and type Seattle.

Press **Enter**.

2 **Save the test.**

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

Running and Analyzing a Parameterized Test

You will now run the Lesson5 test. QuickTest will run the test three times (iterations), once for each departure city in the Data Table. The table below illustrates the data used in each iteration.

Iteration	Departure City
#1	New York
#2	Portland
#3	Seattle

1 **Run the *Lesson5* test.**

▶ Run

Click the **Run** button or choose **Test > Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. When the test run is completed, the Test Results window opens.

2 **Examine the results summary.**

Note that only the first iteration ran successfully.

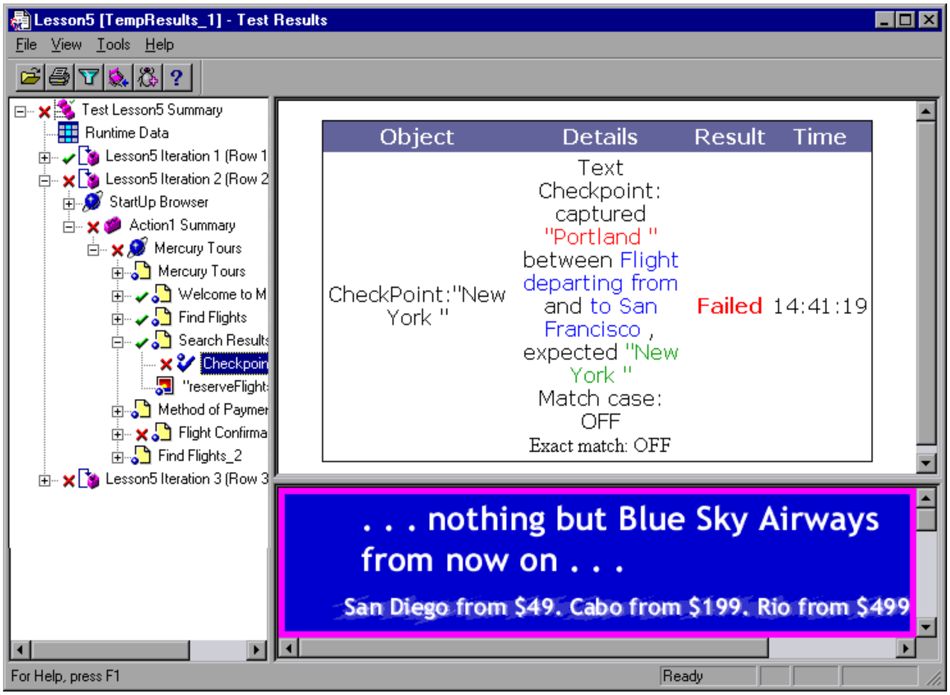
3 **Find the cause of the failed iterations.**

In the **results tree**, expand (+) **Lesson5 Iteration 2 (Row 2)** > **Action1 Summary** > **Mercury Tours** > **Search Results**.

✗

A “failed” mark is displayed next to the Search Results.

Expand (+) **Search Results** > **Checkpoint: “New York”**. Click **Checkpoint: “New York”**.



Object	Details	Result	Time
CheckPoint:"New York "	Text Checkpoint: captured "Portland " between Flight departing from and to San Francisco , expected "New York " Match case: OFF Exact match: OFF	Failed	14:41:19

The text checkpoint failed because QuickTest used Portland as the departure city in Iteration #2. The text checkpoint checks for the constant: "New York". The departure city in Iteration #2 is "Portland". The table below displays the results of the checkpoint in each iteration.

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

Note: The table checkpoint also failed in the second and third iterations, because the ticket price changed when the departure city changed. In Lesson 6, you will modify the table checkpoint so that when you run your test, QuickTest will update the expected ticket price in the checkpoint with the correct ticket price for the current departure city.

4 Exit the Test Results window.

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

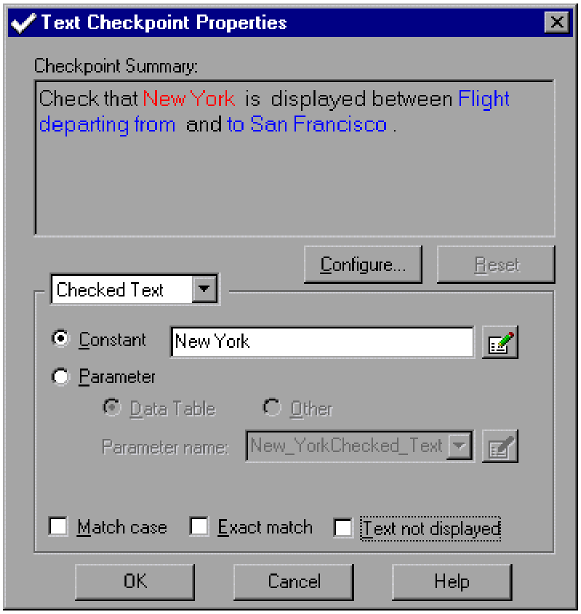
Modifying a Parameterized Test

When you ran your parameterized test, only the first iteration of the text checkpoint passed. The text checkpoint failed in the other two iterations because the text checkpoint did not find the expected text "New York." In this exercise, you will modify your text checkpoint so that when you run your test, QuickTest will update the expected departure city in the checkpoint with the correct departure city of the iteration, based on the information in the Data Table.

1 Locate the text checkpoint to modify.

In the test tree, expand (+) **Search Results** > **Checkpoint: “New York”**.

Right-click **Checkpoint: "New York"** and select **Checkpoint Properties**. The Text Checkpoint Properties dialog box opens.



Text Checkpoint Properties

Checkpoint Summary:

Check that New York is displayed between Flight departing from and to San Francisco .

Checked Text

● Constant: New York

○ Parameter

● Data Table ○ Other

Parameter name: New_YorkChecked_Text

□ Match case □ Exact match □ Text not displayed

2 **Parameterize the text checkpoint.**

In the **Checked Text** box, New York is displayed in the Constant box. New York is the expected value for the checkpoint for each iteration. Click **Parameter**.

Select **departure** from the Parameter name list box. This instructs the checkpoint to use the departure parameter values as the expected results.

Click **OK**. Note that the <*> icon is now displayed beside the checkpoint in Search Results. The icon indicates that the checkpoint is parameterized.

3 **Save the test.**

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

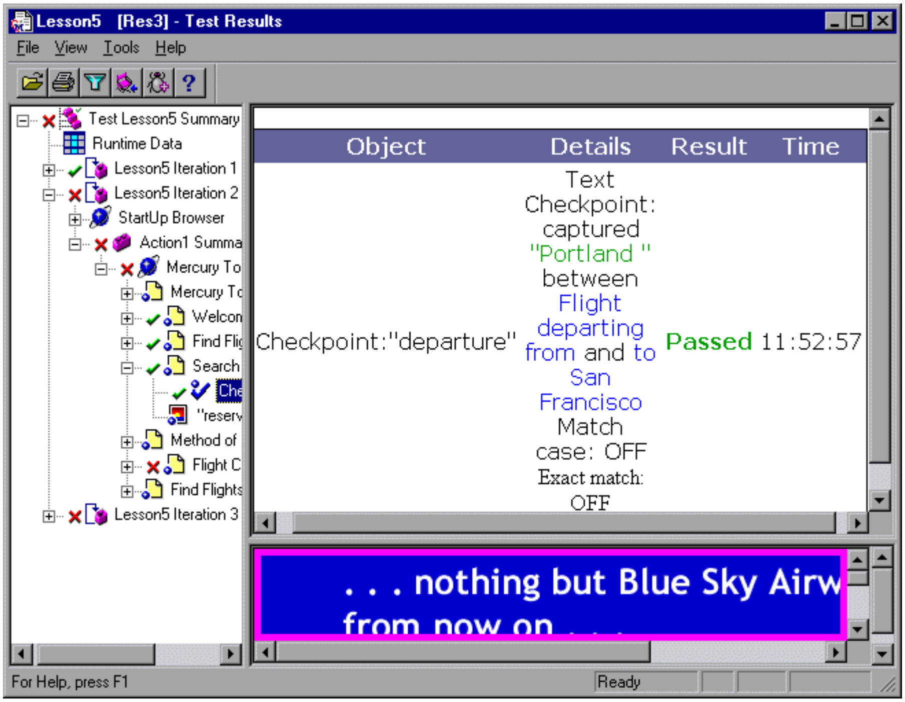
4 Run the modified *Lesson5* test.

▶ Run

Click the **Run** button on the QuickTest toolbar or choose **Test** > **Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. When the test run is completed, the Test Results window opens.

5 Examine the results summary.

In the results tree, expand (+) **Lesson5 Iteration 2** > **Action1 Summary** > **Mercury Tours** > **Search Results** > **Checkpoint:"departure"**. Highlight **Checkpoint:"departure"**.



Object	Details	Result	Time
Checkpoint:"departure"	Text Checkpoint: captured "Portland " between Flight departing from and to San Francisco Match case: OFF Exact match: OFF	Passed	11:52:57

The checkpoint results in the Details pane show that Portland was the expected result and the actual result. Therefore, the checkpoint passed.

6 Repeat step 5 for iteration 3.

The table below displays the results of the text checkpoint in each iteration.

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

Note: The table checkpoint failed in the second and third iterations, because the ticket price changed when the departure city changed. In Lesson 6, you will modify the table checkpoint so that when you run your test, QuickTest will update the expected ticket price in the checkpoint with the correct ticket price for the current departure city.

7 Close the Test Results window.

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

You have now created a parameter that inserts different data into different iterations of a test. In the next lesson you will learn to retrieve data from each iteration.

6

Creating Output Values

In Lesson 5, Parameterizing Tests, you created parameters that inserted different data into each iteration. You can also retrieve data from your Web site in each iteration using output values. QuickTest displays the retrieved data in the Runtime Data Table following the test run. This information can be used later in the test.

For example, the date of the flight is displayed in the Search Results Flights page. You can place the displayed date in an output value. You can then create a text checkpoint in the Flight Confirmation page that checks the date of the flight and uses the date contained in the output value as the expected text. This enables you to verify that the same values are displayed in two different Web pages.

In this lesson you will learn about:

- Creating an Output Value
- Running and Analyzing a Test with Output Values

Creating an Output Value

In this exercise you will create an output value that retrieves the price of the flight from the Method of Payment page for each new departure city in each test iteration. You will then modify the table checkpoint that you created in Lesson 4. The table checkpoint will check that the price displayed in the Flight Confirmation page matches the price captured in the Method of Payment page.

1 **Start QuickTest and open the *Lesson5* test.**

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Open Existing**.

Browse to the *Lesson5* test. Click **Open**.

2 **Save the test as *Lesson6*.**

Select **File** > **Save As**. Save the test as Lesson6.

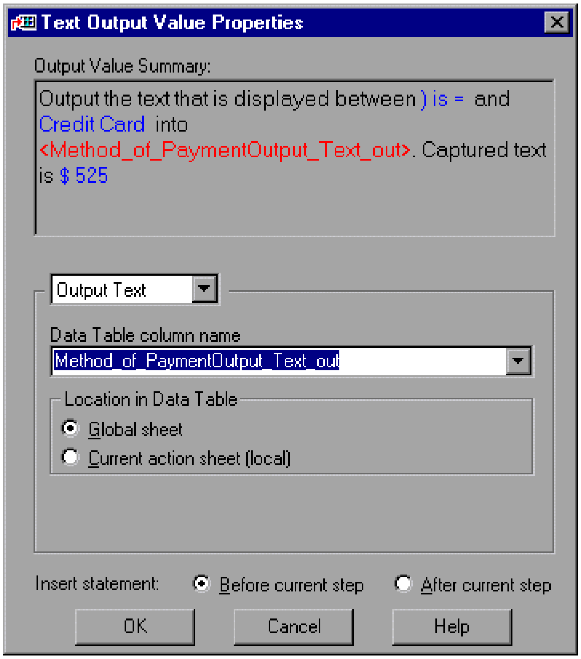
3 **Confirm that the Active Screen option is enabled.**

If you do not see the Active Screen on the right side of the screen, toggle the Active Screen toolbar button or select the Active Screen option from the View menu to enable it. Make sure the Active Screen tab is selected.

4 **Select the text you want to insert into an output value.**

In the test tree, expand (+) **Action 1** > **Mercury Tours** > **Method of Payment**. Highlight the **Method of Payment** page in the test tree. The Method of Payment page is displayed in the Active Screen.

In the Active Screen, scroll to the bottom of the Method of Payment screen (where the ticket price is calculated) and highlight the ticket price ($525). Right-click the highlighted text and select **Insert Text Output**. The Text Output Value Properties dialog box opens.



Text Output Value Properties

Output Value Summary:

Output the text that is displayed between) is = and Credit Card into <Method_of_PaymentOutput_Text_out>. Captured text is $ 525

Output Text

Data Table column name: Method_of_PaymentOutput_Text_out

Location in Data Table

● Global sheet

○ Current action sheet (local)

Insert statement: ● Before current step ○ After current step

5 Set the output value settings.

In the Output Text box, change the **Data Table column name** to ticket_price_output.

The list box contains options for setting the text before and the text after. Accept the Text Before and Text After settings.

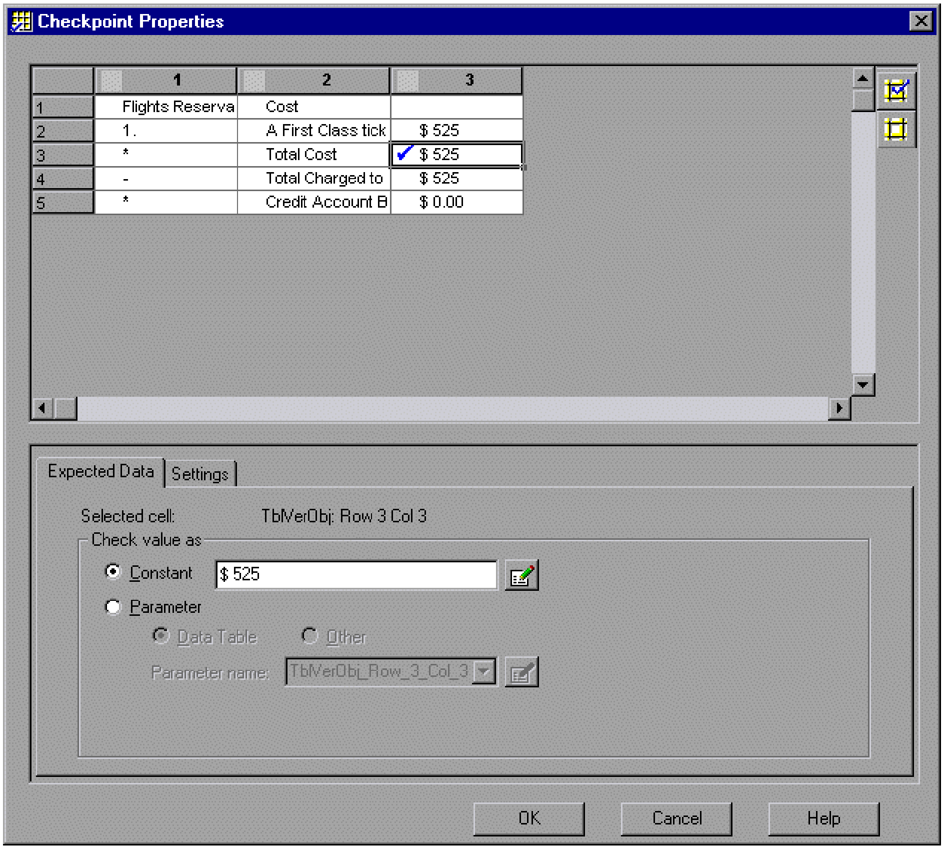
Click **OK**. The ticket_price_output column is added to the Data Table. QuickTest automatically adds $525 as the first value.

QuickTest adds the output value to your test. It is displayed in the test tree as a new step under the Method of Payment page.

6 Modify the table checkpoint for the total cost displayed in the Flight Invoice table.

In the test tree, expand (+) **Action 1** > **Mercury Tours** > **Flight Confirmation** > **Flights Reservation**.

Right-click the table checkpoint, and select **Checkpoint Properties**. The Checkpoint Properties dialog box opens.



	1	2	3
1	Flights Reserva	Cost	
2	1.	A First Class tick	$ 525
3	*	Total Cost	✓ $ 525
4	-	Total Charged to	$ 525
5	*	Credit Account B	$ 0.00

Selected cell: TblVerObj: Row 3 Col 3

Check value as

● Constant $ 525

○ Parameter

● Data Table ○ Other

Parameter name: TblVerObj_Row_3_Col_3

Click cell 3 in row 3. Click **Parameter** and select **ticket_price_output** from the Parameter name list box. Click **OK**. You have parameterized the checkpoint.

7 Save the test.

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

Running and Analyzing a Test with Output Values

You will now run the test and examine the results.

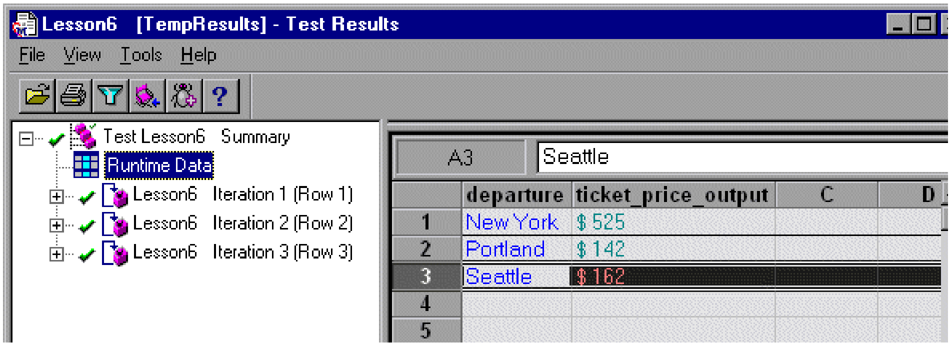
1 Run the *Lesson6* test.

▶ Run

Click the **Run** button or choose **Test** > **Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. When the test run is completed, the Test Results window opens.

2 Examine the runtime data results.

In the Test Results window, select **Runtime Data** from the results tree. The output values used during the test run are displayed in a spreadsheet. Note that a different price is shown in the ticket_price_output column for each iteration.


Lesson6 [TempResults] - Test Results
File
View
Tools
Help
Test Lesson6 Summary
Runtime Data
Lesson6 Iteration 1 (Row 1)
Lesson6 Iteration 2 (Row 2)
Lesson6 Iteration 3 (Row 3)

A3 Seattle

	departure	ticket_price_output	C	D
1	New York	$ 525		
2	Portland	$ 142		
3	Seattle	$ 162		
4				
5				

3 Examine the checkpoint results.

Choose **View** > **Expand All**.

Click the **Checkpoint:"Flights Reservation"** step in Iteration 1.

Note that in the Details pane, the checkpoint is marked in green as "Passed," since the value in the invoice in the Flight Confirmation screen matched the expected value retrieved from the Method of Payment screen.

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

4 Close the Test Results window.

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

7

Using Regular Expressions

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

A regular expression is a string that specifies a complex search phrase.

In this lesson you will learn about:

- Regular Expression Syntax
- Using Regular Expressions
- Running and Analyzing a Test with Regular Expressions

Regular Expression Syntax

QuickTest searches for all characters in a regular expression literally, except for the period (.), asterisk (*), hyphen (-), caret (^), brackets ([ ]), parentheses (()), dollar sign ($), vertical line (|), plus sign (+), question mark (?), and the backslash (\).

When one of these special characters is preceded by a backslash (\), QuickTest searches for the literal character. For example, if you were looking for the Web site called mercurytours.mercuryinteractive.com, the period (.) would be mistaken for an indication of a regular expression. To indicate that this is not the case, you would write it as:

```
mercurytours\.mercuryinteractive\.com.
```

To search for the literal character '\', write '\\' in your expression.

Using Regular Expressions

In this exercise you will verify that the Departure Date in the Search Results page has a valid date format (MM/DD/200Y). To do this, you will create a text checkpoint with a regular expression that will match any single character within a range.

1 Start QuickTest and open the *Lesson2* test.

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Open Existing**.

Browse to the *Lesson2* test. Click **Open**.

2 Save the test as *Lesson7*.

Select **File** > **Save As**. Save the test as Lesson7.

3 Confirm that the Active Screen option is enabled.

If you do not see the Active Screen on the right side of the screen, toggle the Active Screen toolbar button or select the Active Screen option from the View menu to enable it. Make sure the Active Screen tab is selected.

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

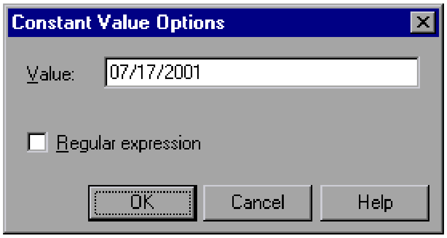
Click **View** > **Expand All** to expand the test tree.

In the test tree, click **Search Results**. The Search Results page is displayed in the Active Screen.

In the Active Screen, highlight the date. Right-click the highlighted string and select **Insert Text Checkpoint**. The Text Checkpoint Properties dialog box opens.

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

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



Constant Value Options

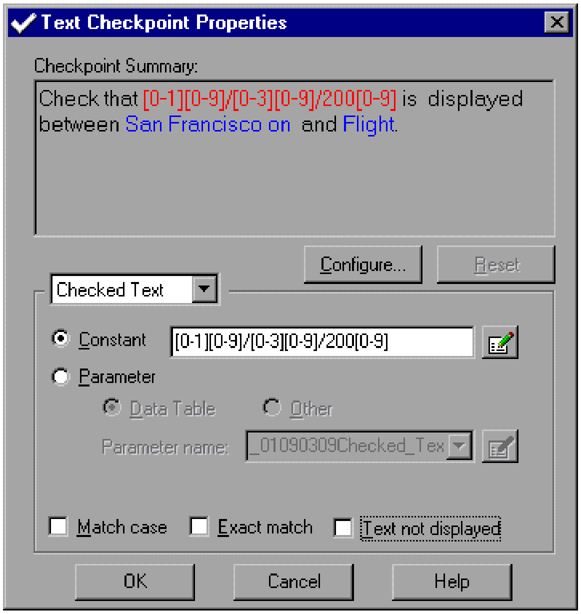
Value: 07/17/2001

☐ Regular expression

Replace the current date in the **Value** box with [0-1][0-9]/[0-3][0-9]/200[0-9]. This instructs QuickTest to check that each character in the selected text matches the number range format defined by the regular expression. The expression checks for the following format: MM/DD/200Y. Select the **Regular expression** check box. A message box prompts you to insert the backslash character before each special character. Click **No**.

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

Click **OK** to close the Constant Value Options dialog box. The Text Checkpoint Properties dialog box is displayed as follows:



Text Checkpoint Properties

Checkpoint Summary:

Check that [0-1][0-9]/[0-3][0-9]/200[0-9] is displayed between San Francisco on and Flight.

Checked Text

● Constant [0-1][0-9]/[0-3][0-9]/200[0-9]

○ Parameter

● Data Table ○ Other

Parameter name: _01090309Checked_Tex

□ Match case □ Exact match □ Text not displayed

Accept the Text Before and Text After settings, and click **OK** to close the Text Checkpoint Properties dialog box.

QuickTest adds the text checkpoint to your test. It is displayed as a new step in the test tree under the Search Results page.

6 Save the test.

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

Running and Analyzing a Test with Regular Expressions

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

1 Run the *Lesson7* test.

Click the **Run** button or choose **Test** > **Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. When the test run is completed, the Test Results window opens.

2 Examine the checkpoint results.

In the results tree, expand (+) **Test Lesson7 Summary** > **Lesson7 Iteration 1 (Row 1)** > **Action1 Summary** > **Mercury Tours** > **Search Results** > **CheckPoint: "[0-1][0-9]/[0-3][0-9]/200[0-9]"**.

Click **CheckPoint:[0-1][0-9]/[0-3][0-9]/200[0-9]**.

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

3 Exit the Test Results window.

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

8

Dividing Tests into Multiple Actions

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

In this lesson you will learn about:

- Dividing Tests into Multiple Actions
- Creating New Actions
- Inserting Existing Actions
- Parameterizing an Action
- Running and Analyzing a Multi-Action Test

Dividing Tests into Multiple Actions

If you were to break down one of the tests you have created in the previous lessons, you would see that it could be divided into several procedures:

- You logged into the Mercury Tours site.
- You chose a departure point and a destination point.
- You selected a flight.
- You paid for the ticket.
- You confirmed your reservation.

Assume that you wanted to run your test using five different departure points. As we saw in Lesson 5, Parameterizing Tests, you could create a data set which would run the test five times using five different sets of data.

You can also organize your test so that only the second procedure would run five times. You can do this by dividing your test into actions.

You divide your test into actions by inserting a new action or by inserting a copy of or a call (link) to an action from another test.

You can insert actions into your test during your recording session or afterwards. You use one of the following menu options or tool bar buttons to insert actions into your test:

- **Insert** > **New Action** or click the **Insert New Action** toolbar button.
- **Insert** > **Copy of Action**
- **Insert** > **Call to Action**

Creating New Actions

In this exercise you will create a test that is divided into three actions.

1 Start QuickTest and open a new test.

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Blank Test**.

If another test is open in QuickTest, choose **File** > **New**. A new test opens.

2 Confirm that the Active Screen option is enabled.

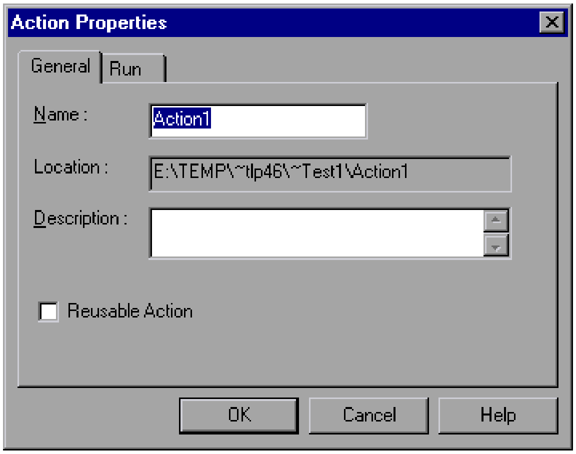
If you do not see the Active Screen on the right side of the screen, toggle the Active Screen toolbar button or select the Active Screen option from the View menu to enable it. Make sure the Active Screen tab is selected.

3 Confirm that automatic page checks will not be added.

Choose **Tools** > **Options** from the menu. In the Web tab, click **Advanced** to open the Advanced Web Options dialog box. Confirm that the **Create a checkpoint for each Web page while recording** check box is cleared. Click **OK** to close the Advanced Web Options dialog box. Click **OK** to close the Options dialog box.

4 Rename the first action in the test to Login.

In the test tree, right-click **Action1** and choose **Action Properties**. The Action Properties dialog box opens.



Name : Action1

Location : E:\TEMP\~tlp46\~Test1\Action1

Description :

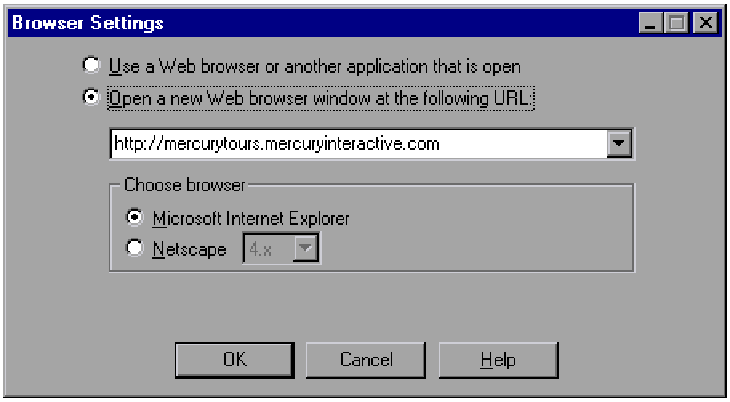
□ Reusable Action

Type Login in the **Name** box. Type Enter user name and password to log into the Mercury Tours Web site in the **Description** box. Click **OK**. The action name is displayed as **Login** instead of **Action1** in the test tree.

5 Start recording on the Mercury Tours Web site.

● Record

Choose **Test** > **Record** or click the **Record** button. The Browser Settings dialog box opens.



○ Use a Web browser or another application that is open

● Open a new Web browser window at the following URL:

http://mercurytours.mercuryinteractive.com

Choose browser

● Microsoft Internet Explorer

○ Netscape 4.x

Accept the default settings and click **OK**. Your browser opens to the Mercury Tours Web site.

6 Log in to Mercury Tours.

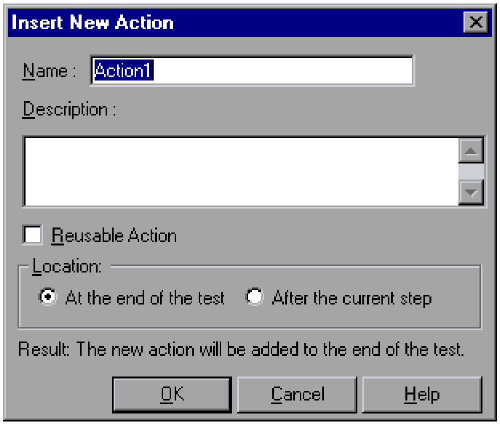
In the **Member name** box, type mercury. In the **Password** box, type mercury. Click **Login**. The Welcome page opens.

Return to QuickTest and maximize its window for the next step. You have just finished creating an action that logs into the site.

7 Create a new action in the test.

Choose **Insert** > **New Action** or click the **Insert New Action** button.

The Insert New Action dialog box opens.



Name : Action1

Description :

☐ Reusable Action

Location:

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

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

In the **Name** box, type Choose Flight and click **OK**. The new action is displayed at the end of your test tree.

Return to the Mercury Tours Web site to continue recording. Ensure that both QuickTest and the Web browser are visible on your screen. Record the following Flight selection steps:

- Click the **Flights** navigation button. The Find Flight page opens.
- In the **Departure City** list, select **New York**. Click **Continue**. The Search Results page opens.
- Select the second flight option and click **Continue**. The Method of Payment page opens.

Return to QuickTest and maximize its window for the next step. You have just finished creating an action that selects the flight.

8 Create the Purchase action.

Repeat the procedure described in step 7, with the following changes:

- Type Purchase as the name of the action.
- When you continue recording, record the following steps:
  - In the Method of Payment page, type 123456789 as your credit card number, and type 1234 as the expiration date. Click **purchase flight**. The Flight Confirmation page opens.
  - From the Flight Confirmation page, click the **sign off** navigation button.

Return to QuickTest and maximize its window for the next step. You have just finished creating an action that enters purchase information. Click the **Stop** button to stop recording.

9 Make the Login action reusable.

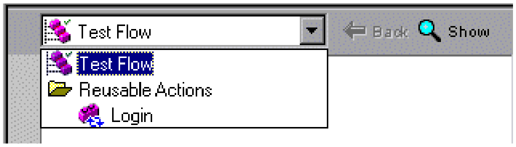
In practice, if you want to call an action from another test, you must make the action a *reusable* action in the test where it is located. In the next part of this lesson, you will call the Login and Purchase actions you just created from another test, so you must make the actions reusable in this test.

Right-click the Login action and choose **Action Properties**. The Action Properties dialog box opens. Select the **Reusable action** check box and click **OK**. Click **Do not show this dialog box again** in the message box that is displayed. Click **OK**. The action icon changes to a reusable action icon .

10 View the details of the Login action.

Note that you no longer have the option to expand the Login action within the test tree.

If the *Action List* was not previously displayed above the test tree, it is displayed after you convert the action to a reusable action. The Action List displays a list of all reusable and external actions in the test. An *external* action is a call to an action from another test.



View the details of the Login action by selecting **Login** from the Action List. Choose **View** > **Expand All**. The details of the Login action are displayed.

In the Action toolbar, click the arrow of the Action List, and select **Test Flow** to return to the main test tree.

11 Make the Purchase action reusable.

Repeat step 9 for the Purchase action.

12 Save the test.

Select **File** > **Save As**. Save the test as Lesson8A.

Inserting Existing Actions

When you plan a suite of tests, you may realize that each test requires one or more identical activities, such as logging in. Rather than creating a new action with identical steps for each test, you can insert either a copy of or a call (link) to an existing action.

When you insert a copy of an existing action, you can make changes to the copied action, and your changes will not affect nor be affected by any other test. Calls to actions, however, are read-only in the calling test. They can only be modified in the test in which they were created. Calls to actions enable you to use the same action in several tests and make it easy to maintain tests, because when your application changes you only have to update the action in the original test.

In this exercise you will create a new test that is similar to Lesson8A, except that the Choose Flight action includes steps that change the seating preference and number of passengers in addition to setting the departure city.

In this exercise you will learn about:

- Inserting a Copy of an Action
- Inserting Calls to Actions
- Modifying an Action

Inserting a Copy of an Action

First, you will insert a copy of the Choose Flight action.

1 Open a new test.

If QuickTest is not already open, choose **Start** > **Programs** > **Astra QuickTest** > **Astra QuickTest**. If the Welcome window opens, click **Blank Test**.

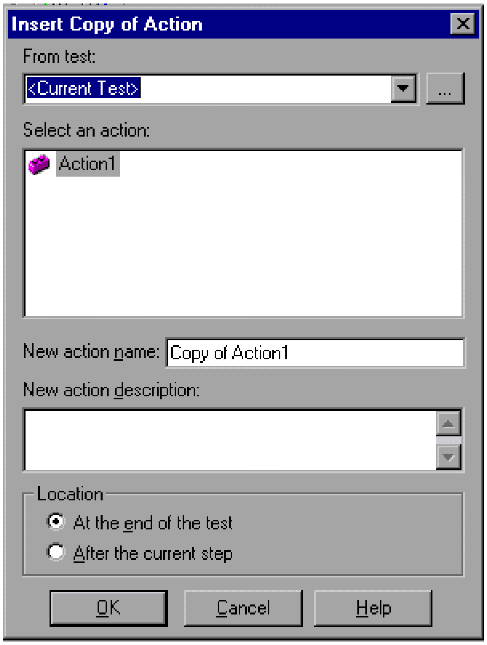
If Lesson8A or another test is open in QuickTest, choose **File** > **New**. A new test opens.

2 Confirm that automatic page checks will not be added.

Choose **Tools** > **Options** from the menu. In the Web tab, click **Advanced** to open the Advanced Web Options dialog box. Confirm that the **Create a checkpoint for each Web page while recording** check box is cleared. Click **OK** to close the Advanced Web Options dialog box. Click **OK** to close the Options dialog box.

3 Insert a copy of the Choose Flight action.

Choose **Insert** > **Copy of Action**. The Insert Copy of Action dialog box opens.



Insert Copy of Action

From test: <Current Test>

Select an action:

Action1

New action name: Copy of Action1

New action description:

Location

● At the end of the test

○ After the current step

Browse to the *Lesson8A* test in the Open QuickTest Test dialog box, and click **Open**. The actions from Lesson8A are displayed in the Insert Copy of Action dialog box. Note that the description of the highlighted action is also displayed.

Highlight the Choose Flight action and click **OK**. The Copy of Choose Flight action is added to the end of the test. Note that the action you inserted is just like any other action in your test. You can modify the steps in the copied action just as you would those of any other action.

Note: You can change the name of the action by right-clicking it and selecting **Action Properties**. In the **Name** box, enter a new name and click **OK**.

---

4 Modify the Copy of Choose Flight action.

Expand **Copy of Choose Flight** > **"Welcome to Mercury"** > **"Find Flights"**. In the test tree, right-click **"depart" Select "New York"** and choose **Method Arguments**. The Method Arguments dialog box opens.

Change the value in the **Constant** box to London. Click **OK**.

5 Delete Action1 from the test.

Every test opens with a default Action1. Because each test must always have at least one action, it was necessary to insert the Choose Flight action before deleting Action 1. Now you can delete Action1.

Highlight Action1 and choose **Edit** > **Delete** or press the **Delete** key on your keyboard. A confirmation message opens. Click **Yes** to confirm.

6 Save the test.

Select **File** > **Save As**. Save the test as Lesson8B.

7 Confirm that the original Choose Flight action was not modified.

Choose **File** > **Open**. Browse to *Lesson8A*, and click **Open**.

Expand **Choose Flight** > **"Welcome to Mercury"** > **"Find Flights"**. Note that the modification made in step 4 has not affected the original action.

Inserting Calls to Actions

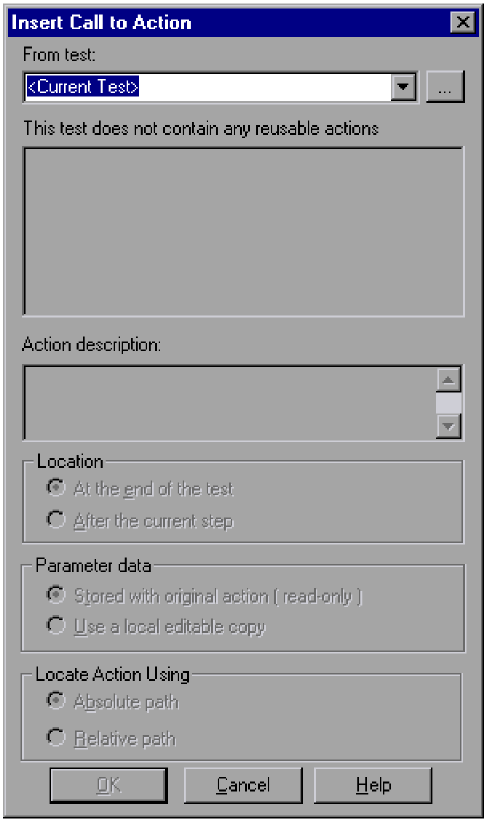
Now you will insert calls to the Login and Purchase actions, which are in Lesson8A.

1 Open the *Lesson8B* test.

Choose **File** > **Open**. Browse to *Lesson8B*, and click **Open**.

2 Insert a call to the Login action.

Choose **Insert** > **Call to Action**. The Insert Call to Action dialog box opens.



Browse to the *Lesson8A* test in the Open QuickTest Test dialog box, and click **Open**. Note that only reusable actions from Lesson8A are displayed in the Insert Call to Action dialog box.

Highlight the Login action and click **OK**. The Login action is added to the end of the test. The Login action is displayed as an *external* action with the icon in the test tree.

Note that you do not have the option to expand the Login action in the test tree. You can view the Login action using the Action List. Note that the action you inserted is read-only. To modify the action, you must open the original test and modify the action there.

3 Move the Copy of Choose Flight action below the Login action.

The Copy of Choose Flight action should be the second action in your test.

Highlight the Copy of Choose Flight action and drag it below the Login action.

4 Insert a call to the Purchase action.

Use the instructions in step 2 to insert a call to the Purchase action in Lesson8A.

5 Save the test.

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

6 Modify the original Purchase action.

Choose **File** > **Open**. Browse to *Lesson8A*, and click **Open**.

Click the arrow in the Action List and select the **Purchase** action. Expand **Purchase** > **"Method of Payment"** > **"Method of Payment"**. In the test tree, right-click **"expDate" Set "1234"** and choose **Method Arguments**. The Method Arguments dialog box opens.

Change the value in the **Constant** box to 4321. Click **OK**.

7 Save the test.

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

8 Confirm that the modification made in the original Purchase action is also in the calling test.

Choose **File** > **Open**. Browse to *Lesson8B*, and click **Open**.

Click the arrow in the Action List and select the **Purchase[Lesson8A]** action. Expand **Purchase[Lesson8A]** > **"Method of Payment"** > **"Method of Payment"**. Note that the modification made in step 5 to the original action exists also in the calling test.

In the Action toolbar, click the arrow of the Action List, and select **Test Flow** to return to the main test tree for the next section.

Modifying an Action

In the previous section, you inserted the Choose Flight action as a copy rather than a call because you need to change it. You want to select additional steps to the Find Flights page.

1 Set the Mercury Tours Web application to the right location.

If it is not already open, open Internet Explorer to *http://mercurytours.mercuryinteractive.com*. Type mercury in the **Member name** and **Password** boxes and click **Login**. On the next page, click **Flights** to open the Find Flights page. Select **London** as the Departure City. Ensure that QuickTest and the Web browser are both visible.

2 Choose the step after which you want to begin recording.

In QuickTest, choose **View** > **Expand All** to view all the steps of the Copy of Choose Flight action.

Highlight the "depart" Select "London" step.

3 Start recording on the open Mercury Tours Web site.

Choose **Test** > **Record** or click the **Record** button. The Browser Settings dialog box opens. Select **Use a Web browser or another application that is open**. Click **OK**. QuickTest will record new steps directly after the "depart" Select "London" step. Ensure that both QuickTest and the Find Flight Web page are visible on screen.

4 Add the new steps.

In the Find Flight Web page, type 3 in the **No. of Passengers** box and select **Aisle** in the **Seating Preference** option.

Click **Stop** in QuickTest.

5 Fix the startup test settings.

When you selected **Use a Web browser or another application that is open** in the Browser Settings dialog box in step 3, that also changed the startup test settings for the test. When you run the test, however, you want the test to launch a new browser and load the Mercury Tours home page.

Choose **Test** > **Settings**.

In the **Browser** tab, select **Open a new Web browser window at the following URL**.

Confirm that the URL is: *http://mercurytours.mercuryinteractive.com.*

Click **OK** to save the changes and close the Test Settings dialog box.

6 Save the test.

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

Parameterizing an Action

If you look at the Data Table at the bottom of the QuickTest screen in the Lesson8B test, you will see that four tabs exist: Global, Copy of Choose Flight, Login [Lesson8A], and Purchase [Lesson8A]. The Global tab represents a data sheet with data that will be used for the entire test. If five sets of data are displayed in the Global table, the test will run five times. In addition, you can create data sets for each action, using the relevant *action* sheet. If you create a data set containing five sets of data within an action, that action will run five times within one test iteration. In this section, you will create a data set within the Choose Flight action.

1 Open the *Lesson8B* test.

Choose **File** > **Open** and browse to *Lesson8B*. Click **Open**.

2 Select the step you want to parameterize.

Expand **(+) Copy of Choose Flight** > **“Welcome to Mercury”** > "**Find Flights**" > "**depart**" **Select** "**London**".

Right-click "depart" Select "London".

Choose **Method Arguments**. The Method Arguments dialog box opens.

3 Enter the parameter information.

In the Edit value box, select **Parameter**.

Give the parameter a logical name. In the **Parameter name** box, type Departure.

You are creating this data set exclusively for the Copy of Choose Flight action. A parameter for an action is called an *action* parameter. Confirm that **Data Table** is selected. Click the **Edit Parameter Options** button to open the Data Table Parameter Options dialog box. Click **Current action sheet (local)**. Click **OK** to close the Data Table Parameter Options dialog box.

Click **OK** to close the Method Arguments dialog box. QuickTest automatically adds the Departure parameter to the Copy of Choose Flight data sheet and inserts London into the first cell in the Departure column.

4 Enter the parameterization data in the Data Table.

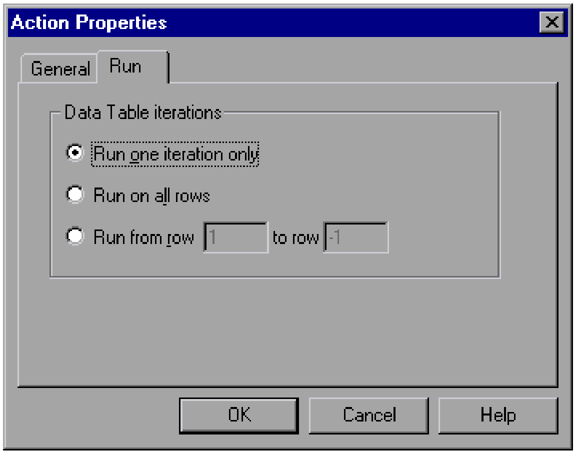
In the Data Table, click the **Copy of Choose Flight** tab. Type Portland in the second row of the Departure column. Press **Enter**.

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

By default, QuickTest will only run the data that is displayed in the first row of the local sheet.

Right-click the **Copy of Choose Flight** action in the test tree and choose **Action Properties**. The Action Properties dialog box opens.

Click the **Run** tab.



Data Table iterations

● Run one iteration only

○ Run on all rows

○ Run from row 1 to row -1

You want the action to use all the data in the Data Table. Select **Run on all rows**. Click **OK** to close the message box that is displayed. Click **OK** to close the Action Properties dialog box.

6 Save the test.

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

Running and Analyzing a Multi-Action Test

You will now run the Lesson8B test. The entire test will run once, but the Choose Flight action will run twice; one time for each set of data in the Choose Flight data sheet.

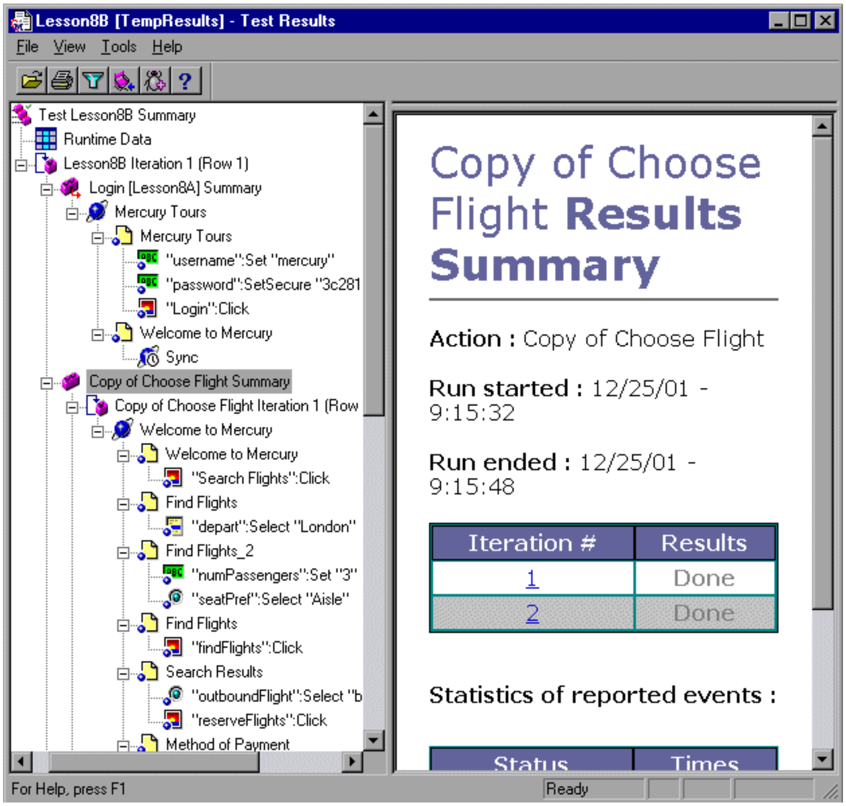
1 Run the *Lesson8B* test.

Click the **Run** button or choose **Test** > **Run**. The Run Results Destination dialog box opens. Select **Temporary run results folder**, and click **OK**. When the test run is completed, the Test Results window opens.

2 Examine the test results.

Choose **View** > **Expand All** in the Test Results window. Examine the Results Summary. Iteration 1 (the only iteration in the test) is marked as Done. This indicates that the test ran without any failures. The statistics below it summarize the warnings, passes and failures that occurred from checkpoints and report messages during the test.

Highlight the **Copy of Choose Flight Summary** action in the Results Tree. Note that within the Copy of Choose Flight Results Summary, the results of both iterations are displayed.



Copy of Choose Flight **Results Summary**

Action : Copy of Choose Flight

Run started : 12/25/01 - 9:15:32

Run ended : 12/25/01 - 9:15:48

Iteration #	Results
1	Done
2	Done

Statistics of reported events :

Status	Times

3 Close the Test Results window.

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

9

Where Do You Go from Here?

Now that you have completed the exercises in this tutorial, you are ready to apply the QuickTest concept and skills you learned to your own Web site.

In this lesson you will learn about:

- Testing Your Own Web Site—Getting Started
- QuickTest Features
- Getting Additional Information

Testing Your Own Web Site—Getting Started

In this tutorial you learned the basic tools you need to test your own Web site. The list below illustrates the test flow in QuickTest. Use this test flow to create your test.

1 **Plan your test.**

Decide how to organize your test. Consider what users will want to accomplish on your Web site in order to decide which operations to record and which tests to create. Consider whether to divide your test into *actions*, (described in Lesson 8, Dividing Tests into Multiple Actions) and if so, what kind.

2 **Record your test on your Web site.**

While recording (described in Lesson 2, Recording Tests), follow the steps you expect users to perform as they navigate through your Web site.

3 **Enhance your recorded test.**

- Add *checkpoints* (described in Lesson 4, Creating Checkpoints), to search for specific values of a page, object, text string, or table.
- Replace fixed values in your test with *parameters* (described in Lesson 5, Parameterizing Tests), to check how your Web site performs the same operations with multiple sets of data.
- Create *output values* (described in Lesson 6, Creating Output Values), to retrieve data from your Web site during a test run.
- Add *regular expressions* (described in Lesson 7, Using Regular Expressions), to increase the flexibility and adaptability of your tests.
- Use programming and conditional statements to add logic to your test. For additional information, refer to the *Astra QuickTest User's Guide*.

4 **Debug your test.**

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

5 **Run your test on your Web site.**

Run your test on your Web site (described in Lesson 3, Running and Analyzing Tests) to check that it behaves as expected.

6 **Analyze the test results.**

Examine the test results (described throughout this tutorial) to pinpoint defects in your Web site.

7 **Report defects.**

If you have TestDirector installed, you can report any defects to a database. TestDirector is Mercury Interactive's software test management tool. For additional information, refer to the *Astra QuickTest User's Guide* and the documentation included in TestDirector.

QuickTest Features

This tutorial includes lessons on the most commonly used QuickTest features. There are, however, many features that are not described in this tutorial. For example, in QuickTest, you can:

- record and run tests on ActiveX controls. If you have QuickTest Professional, you can also record and run tests on Java applets and applications, Macromedia Flash controls, and RealPlayer and Windows Media Player applications and controls.
- use the *Object Spy* to view the properties of any object in an open application.
- view a list of all objects in an action, and their property values in the *Object Repository* dialog box.
- create *database checkpoints* that enable you to check values in databases accessed by your Web site.
- use Excel formulas in the *Data Table* to create contextually relevant data during the test run.
- use the *Exception Editor* to handle unexpected events and errors that occur during the test run.
- modify how QuickTest identifies objects, including *dynamically changing* objects.
- debug your test by running it in *steps*, and then view the values of variables at *breakpoints* in the *Debug Viewer.*
- use the *Test Batch Runner* to run several tests in succession.
- *configure* the *events* you want QuickTest to record for each type of Web object.
- enhance your test with programming by adding methods, conditional statements, and comments to your test and by sending messages to your test results.

- create, view, and edit your test in the *Expert View*, using VBScript.
- customize how the Expert View is displayed.
- set global and test-specific testing options through the user interface.
- set global and test-specific testing options through the test script in Expert View.

For additional information about these and other features, refer to the *Astra QuickTest User's Guide*.

Getting Additional Information

For more information on Astra QuickTest, refer to the printed documentation and online resources provided with Astra QuickTest.

Printed Documentation

In addition to this tutorial, Astra QuickTest comes with the following printed documentation:

Astra QuickTest Installation Guide explains how to install Astra QuickTest on a single computer or on a network.

Astra QuickTest User's Guide provides step-by-step instructions on how to use Astra QuickTest to test your Web site. It describes many useful testing tasks and options not covered in this tutorial.

Astra QuickTest Object Model Reference provides access to the Astra QuickTest VBScript methods, including a description of each object, a list of the methods associated with each object, description syntax, and an example of usage for each object and method.

Astra QuickTest Shortcut Key Reference Card provides a list of commands that you can execute using shortcut keys.

Online Resources

In addition to this tutorial, QuickTest comes with the following online documentation:

ReadMe (available from the Astra QuickTest Start menu program group) provides the latest news and information about Astra QuickTest.

What's New in Astra QuickTest (available from the Astra QuickTest Help menu) describes the newest features and supported environments in the latest versions of Astra QuickTest.

Books Online (available from the Astra QuickTest Start menu program group or the Astra QuickTest Help menu) displays context-sensitive help for dialog boxes, the *Astra QuickTest Object Model Reference*, and the *Astra QuickTest User's Guide* in an HTML Help browser.

Astra QuickTest Context-Sensitive Help (available from dialog boxes and windows) describes QuickTest dialog boxes and windows.

Astra QuickTest Object Model Reference (available from the Help menu) gives you online access to the VBScript methods, including a description of each object, a list of the methods associated with each object, and description syntax and an example of usage for each object and method.

Mercury Tours sample Web site (available from the Astra QuickTest Browser Settings dialog box) is the basis for this tutorial. The URL for this Web site is *http://mercurytours.mercuryinteractive.com*.

VBScript Reference (available from the Astra QuickTest Help menu) describes Microsoft's VBScript language, and includes a tutorial and function reference.

Technical Support Online (available from the Astra QuickTest Help menu) uses your default Web browser to open Mercury Interactive's Customer Support Web site. This site enables you to browse the knowledge base and add your own articles, post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. The URL for this Web site is *http://support.mercuryinteractive.com*.

Support Information (available from the Astra QuickTest Help menu) presents Mercury Interactive's Web site and Customer Support site, information on user discussion forums, and the location of Mercury Interactive's worldwide offices.

Mercury Interactive on the Web (available from the Astra QuickTest Help menu) uses your default Web browser to open Mercury Interactive's home page. This site provides you with the most up-to-date information on Mercury Interactive and its products. This includes new software releases, seminars and trade shows, customer support, educational services, and more. The URL for this Web site is *http://www.mercuryinteractive.com*.

Index

MERCURY INTERACTIVE

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089 USA

Main Telephone: (408) 822-5200
Sales & Information: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (408) 822-5300

Home Page: www.mercuryinteractive.com
Customer Support: support.mercuryinteractive.com