

Astra® LoadTest™
Virtual User Recorder User's Guide
Version 3.0



MERCURY INTERACTIVE

Astra LoadTest User's Guide, Version 3.0

© Copyright 2000 by Mercury Interactive Corporation

All rights reserved. All text and figures included in this publication are the exclusive property of Mercury Interactive Corporation, and may not be copied, reproduced, or used in any way without the express permission in writing of Mercury Interactive. Information in this document is subject to change without notice and does not represent a commitment on the part of Mercury Interactive.

Mercury Interactive may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents except as expressly provided in any written license agreement from Mercury Interactive.

WinRunner, XRunner, LoadRunner, TestDirector, TestSuite, WebTest, and Astra are registered trademarks of Mercury Interactive Corporation in the United States and/or other countries. Astra, Astra SiteManager, Astra SiteTest, Astra QuickTest, Astra LoadTest, Topaz, RapidTest, QuickTest, Visual Testing, Action Tracker, Link Doctor, Change Viewer, Dynamic Scan, Fast Scan, and Visual Web Display are trademarks of Mercury Interactive Corporation in the United States and/or other countries.

This document also contains registered trademarks, trademarks and service marks that are owned by their respective companies or organizations. Mercury Interactive Corporation disclaims any responsibility for specifying which marks are owned by which companies or organizations.

If you have any comments or suggestions regarding this document, please send them via e-mail to documentation@mercury.co.il.

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089 USA

Table of Contents

Welcome to Astra LoadTest	vii
Using This Guide	vii
Additional Resources	ix
Chapter 1: Introduction	1
Testing with Astra LoadTest	1
Astra LoadTest Testing Process	2
Testing in Expert View Mode	4
Sample Application	4
Chapter 2: The Virtual User Recorder at a Glance.....	5
Starting the Virtual User Recorder	6
The Virtual User Recorder Window	6
Test Pane.....	8
Display Pane	9
Data Pane.....	10
Using Astra LoadTest Commands	10
Chapter 3: Creating Tests	15
About Creating Tests	15
Planning a Test	16
Recording a Test	16
Understanding Your Test	18
Managing a Test	20
Chapter 4: Creating Checkpoints.....	21
About Creating Checkpoints.....	21
Adding a Page Checkpoint	23
Adding a Text or Object Checkpoint	26
Checking Text	29
Checking Objects	32
Modifying Checkpoints.....	34

Chapter 5: Testing Load	35
About Transactions.....	35
About Rendezvous Points.....	37
About Setting Run-time Options.....	38
Setting Run-time Options	38
Run-time Settings	39
Sending Messages to Output	47
Chapter 6: Parameterizing Tests	49
About Parameterizing Tests.....	49
Parameterizing Steps	51
Parameterizing Checkpoints	56
Editing the Data Table.....	58
Chapter 7: Creating Output Parameters	63
About Creating Output Parameters.....	63
Creating Output Parameters.....	64
Parameterizing Text Output.....	68
Parameterizing Object Output	70
Parameterizing Table Output	72
Chapter 8: Using Regular Expressions	73
About Regular Expressions	73
Using Regular Expressions in Steps.....	74
Using Regular Expressions in Object Checkpoints.....	78
Using Regular Expressions in Text Checkpoints.....	80
Regular Expression Syntax	82
Chapter 9: Testing in Expert View	85
About Testing in Expert View	85
Understanding Your Test Script Statements	86
Programming in the Expert View.....	87
Modifying Your Test Script	91
Chapter 10: Enhancing Your Tests with Programming	93
About Enhancing Your Tests with Programming.....	93
Inserting Functions	94
Using Conditional Statements	97
Sending Messages to Your Test Results	101
Adding Comments	102

Chapter 11: Running Tests in Stand-Alone Mode	103
About Running a Test.....	103
Running a Test.....	104
Viewing the Results of a Test Run.....	104
Printing Test Results.....	107
Chapter 12: Debugging Tests	109
About Debugging Test Scripts	109
Using the Step Commands.....	110
Pausing Test Runs.....	111
Setting Breakpoints.....	111
Deleting Breakpoints.....	112
Chapter 13: Setting Testing Options	113
About Setting Testing Options.....	113
Setting Testing Options from the Test Settings Dialog Box.....	114
Testing Options	115
Index	121

Welcome to Astra LoadTest

Welcome to Astra LoadTest, Mercury Interactive's load testing tool for Web applications. Astra LoadTest provides everything you need to quickly create and run tests.

Using This Guide

This guide describes how to use Astra LoadTest to test your Web application. It contains 13 chapters:

Chapter 1 Introduction

Provides an overview of testing with Astra LoadTest.

Chapter 2 The Virtual User Recorder at a Glance

Explains how to start the Virtual User Recorder and introduces the Virtual User Recorder window.

Chapter 3 Creating Tests

Describes how to create and manage tests.

Chapter 4 Creating Checkpoints

Describes how to create checkpoints that enable you to check your application's behavior.

Chapter 5 Testing Load

Describes how to customize your script to conduct accurate load testing through the use of transaction and rendezvous points.

Chapter 6 Parameterizing Tests

Describes how to expand the scope of a basic test by replacing fixed values with parameters, and running the test in iterations.

Chapter 7 Creating Output Parameters

Describes how to retrieve the value of a parameterized argument and assign it as an output parameter into your data table.

Chapter 8 Using Regular Expressions

Describes how to use regular expressions to identify Web objects and text strings with varying values.

Chapter 9 Testing in Expert View

Describes how to enhance your test using the Expert View tab. This chapter is recommended for advanced users of Astra LoadTest.

Chapter 10 Enhancing Your Tests with Programming

Describes how to enhance your test using a few programming techniques.

Chapter 11 Running Tests in Stand-Alone Mode

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

Chapter 12 Debugging Tests

Describes how to control test runs to identify and isolate defects in tests.

Chapter 13 Setting Testing Options

Describes how to change system defaults to adapt Astra LoadTest to your testing environment.

Additional Resources

Astra LoadTest includes the following online resources:



Read Me First provides last-minute news and information about Astra LoadTest.

Astra LoadTest Context Sensitive Help describes dialog boxes and toolbar buttons.

Astra LoadTest Tutorial teaches you basic Astra LoadTest skills and shows you how to start load testing your Web application.

Astra LoadTest QuickTour provides a basic overview of load testing with Astra LoadTest.

Technical Support Online uses your default Web browser to open Mercury Interactive's Customer Support web site.

Support Information presents the locations of Mercury Interactive's Customer Support web site and home page, the e-mail address for sending information requests, the name of the relevant news group, the location of Mercury Interactive's public FTP site, and a list of Mercury Interactive's offices around the world.

Mercury Interactive on the Web 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.

1

Introduction

Welcome to Astra LoadTest, Mercury Interactive's load testing tool for Web applications. This guide provides you with detailed descriptions of Astra LoadTest features and testing procedures.

Testing with Astra LoadTest

Astra LoadTest facilitates creating tests on your Web application by recording as you navigate. You record your test with the Virtual User Recorder. As you navigate through your application, the Virtual User Recorder records each action you perform and generates a test that graphically displays each action as an icon-based step in a *test tree*. For example, actions such as clicking a link, selecting a check box, or submitting a form are recorded in your test.

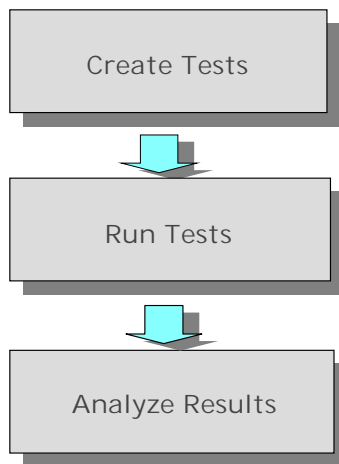
In addition, you can instruct Astra LoadTest to check the response of your application to specific Web objects, text strings, or tables. For example, you can instruct Astra LoadTest to check that a specific text string appears in a particular location on your Web page, or you can check that a hypertext link goes to the correct URL address.

After you record, you can further enhance your test by adding and modifying steps in the test tree. When you run the test, Astra LoadTest connects to your application and performs each step in your test. After you run your test, you can view a report detailing which steps in your test succeeded or failed.

Once you check the functionality of your test in the Virtual User Recorder, you incorporate it in a load testing scenario. You use the Astra Load Test Controller to run load tests and analyze your Web application's performance under load. Refer to the *Astra LoadTest Controller User's Guide* for information about load testing scenarios.

Astra LoadTest Testing Process

Testing with Astra LoadTest involves 3 main stages:



Creating Tests

You create a test by recording a Web session with the Virtual User Recorder. The test is used to load test your application.

To create a test:

- Record a session on your application.

As you navigate through your application, the Virtual User Recorder graphically displays each *step* you perform in the form of a collapsible icon-based *test tree*. A step is any action that changes the content of a Web page in your application, for example, clicking a link or image, or submitting a form of data. For more information, see Chapter 3, “Creating Tests.”

- Insert checkpoints into your test.

A *checkpoint* searches for a specific value of an object or a text string on a Web page and enables you to identify whether or not your Web application is functioning correctly. You can check a Web page for objects, text strings, and tables. For more information, see Chapter 4, “Creating Checkpoints.”

- Insert load testing elements into your test.

You define *transactions* to mark the business processes that Astra LoadTest should measure. When you record a test, Astra LoadTest automatically marks each step you perform as a transaction. This means that when you run a load testing scenario, each step in your test tree is recognized as a transaction to be measured.

You insert *rendezvous points* into a test to emulate heavy user load on the server. Rendezvous points instruct Virtual Users (Vusers) to wait during test execution for multiple Vusers to arrive at a certain point or meeting place. Once all the Vusers arrive at the rendezvous point, they are released and simultaneously perform a task.

- Broaden the scope of your test by replacing fixed values with parameters.

When you test your application, you can *parameterize* your test to check how your Web application performs the same operations with multiple sets of data. The data is stored in a *data table*. When you parameterize your test, Astra LoadTest substitutes the parameters in your test with values from the data table. During each iteration of your test, Astra LoadTest changes the values in the parameterized statements. For more information, see Chapter 6, “Parameterizing Tests.”

Running Tests

After you create your test, you run it using the Virtual User Recorder to debug it before you incorporate it into a load testing scenario.

To run a test:

- Run your test to check your application.

The test runs from the first line in your test and stops at the end of the test. While running, Astra LoadTest connects to your Web application and performs each operation in your test, checking any text strings, objects or tables you specified. If you parameterized your test, Astra LoadTest repeats

the test for each set of data values you defined. For more information, see Chapter 11, “Running Tests in Stand-Alone Mode.”

► Run a test to debug your test.

You can control your test run to help you identify and eliminate defects in your test. You can use the *Step* commands to run a single step in your test. You can also set *breakpoints* to pause your test at pre-determined points. For more information, see Chapter 12, “Debugging Tests.”

Analyzing Test Results

After you run your test, you can view the test results.

To analyze test results:

► View the test results in the Test Results window.

After you run your test, the Test Results window opens and displays the results of your test. You can view a summary of your test results or a detailed report. For more information, see “Viewing the Results of a Test Run”, on page 104.

Testing in Expert View Mode

If you are an advanced user, you can use the Expert View tab to view a text-based version of your test. The test script is composed of VBScript statements (Microsoft’s Visual Basic Scripting language), which correspond to the steps and checks displayed in your test tree. For more information, see Chapter 9, “Testing in Expert View.”

Sample Application

Many examples in this guide use the Mercury Tours sample Web application. The URL for this Web site is <http://astra.mercur-int.com/mercurytours>.

The first page of the Mercury Tours application is the login page. You must log in to start the application. To log in, enter “mercury” as your member name and “mercury” as your password.

2

The Virtual User Recorder at a Glance

This chapter explains how to start the the Virtual User Recorder and introduces the the Virtual User Recorder window.

This chapter describes:

- Starting the Virtual User Recorder
- The Virtual User Recorder Window
- Test Pane
- Display Pane
- Data Pane
- Using the Virtual User Recorder Commands

Starting the Virtual User Recorder



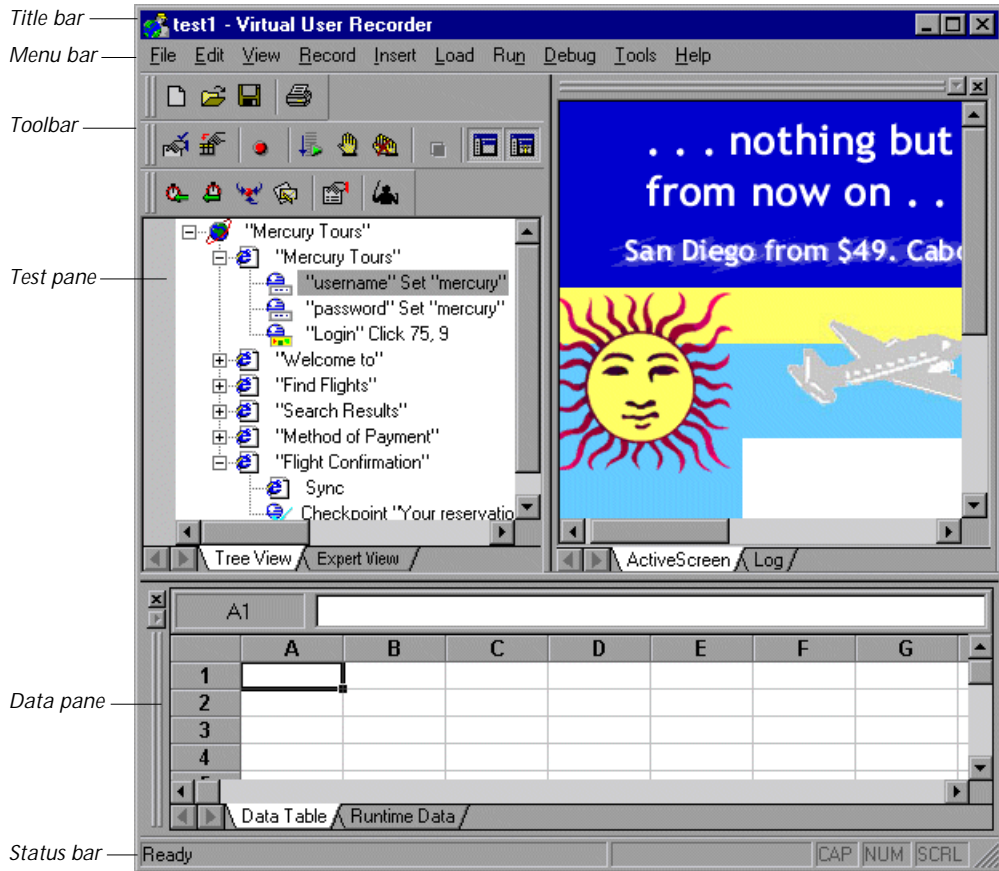
To start the Virtual User Recorder, click **Programs > Astra LoadTest > Products > Virtual User Recorder** in the Start menu. After several seconds, the Virtual User Recorder window is displayed on your desktop.

The Virtual User Recorder Window

The Virtual User Recorder window contains the following key elements:

- ▶ *The Virtual User Recorder title bar*, displaying the name of the currently open test
- ▶ *Menu bar*, displaying menus of the Virtual User Recorder commands
- ▶ *Toolbar*, containing buttons of the most frequently used commands
- ▶ *Test pane*, containing two tabs to view your test—Tree View and Expert View
- ▶ *Display pane*, containing two tabs to assist you in the testing process—ActiveScreen and log
- ▶ *Data pane*, containing two tabs to assist you in parameterizing your test—Data Table and Runtime Data

► *Status bar*, displaying the status of the open test



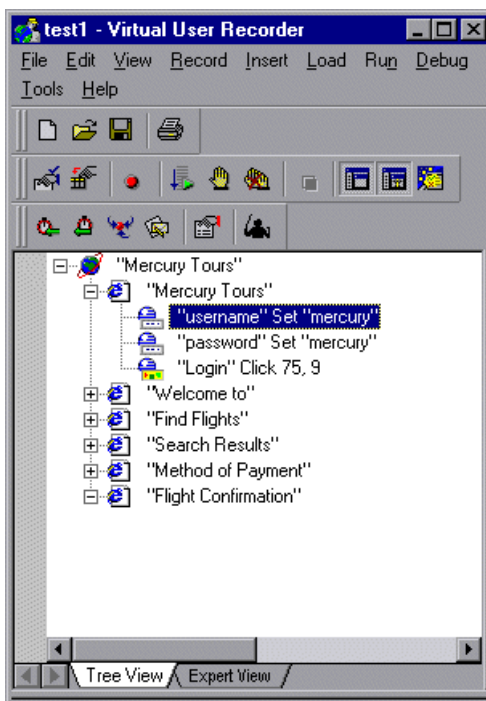
Display pane

Test Pane

The Test pane contains two tabs to view your test—Tree View and Expert View.

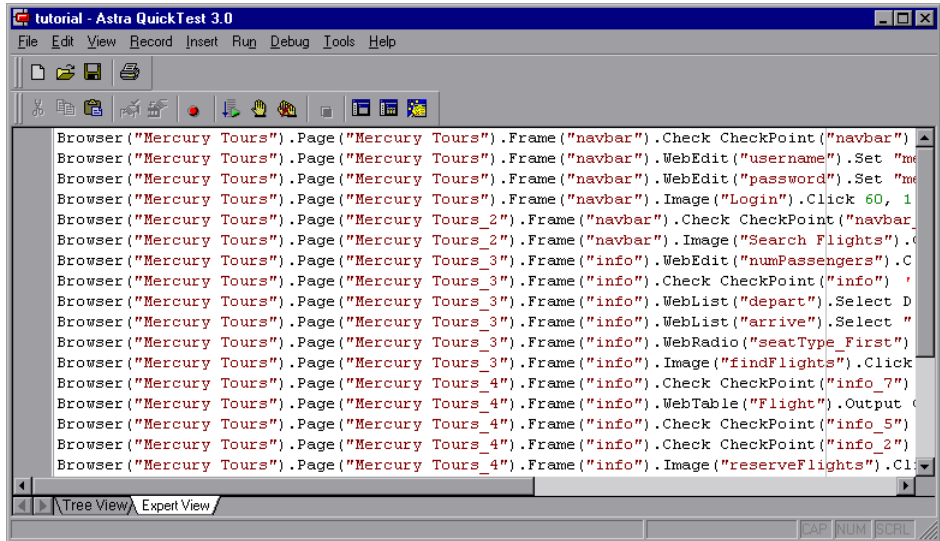
Tree View Tab

In the Tree View tab (default mode), the Virtual User Recorder displays your test in the form of a collapsible icon-based test tree. Each action performed on a your Web application is recorded as an icon-based step in your test tree. For every icon in the Tree View, the Virtual User Recorder displays a corresponding line of script in the Expert View.



Expert View Tab

In the Expert View tab, the Virtual User Recorder displays your test in the form of a test script instead of a test tree. Your test script is composed using the VBScript language. For every statement in the Expert View tab, a corresponding icon exists in the test tree in the Tree View tab. For more information on using the Expert View, see Chapter 9, “Testing in Expert View.”



Display Pane



The Virtual User Recorder’s Display pane contains two tabs to assist you in the testing process—ActiveScreen and Log. To view this pane, click the Display Views button or choose View > Display Views.

ActiveScreen Tab

The ActiveScreen tab displays the Web page or object corresponding to a highlighted step in your test. It provides you with an easy way to view your test, make modifications, and add checkpoints.

Log Tab

The Log tab displays an audit trail of your test run. It lists all the operations performed while running your test.

Data Pane



The Data pane contains two tabs to assist you in the parameterizing your test—Data Table and Runtime Data. To view this pane, click the Data Views button or choose View > Data Views.

Data Table Tab

The Data Table tab contains the variable values for the parameters defined when you create a parameterized test. When you run your parameterized test, Astra LoadTest reads the data from the data table into the test.

Runtime Data Tab

During a test run, the read-only Runtime Data tab displays the actual variable values retrieved from your application and used in your parameterized test.

Using Astra LoadTest Commands

You can select the Virtual User Recorder commands from the menu bar or from a toolbar. Certain Virtual User Recorder commands can also be executed by pressing shortcut keys.

Choosing Commands on a Menu

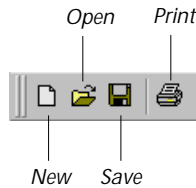
You can choose all Virtual User Recorder commands from the menu bar.

Clicking Commands on a Toolbar

You can execute some Virtual User Recorder commands by clicking buttons on the toolbars. The Virtual User Recorder has three built-in toolbars: the *Standard toolbar*, *Test toolbar*, and the *Load toolbar*.

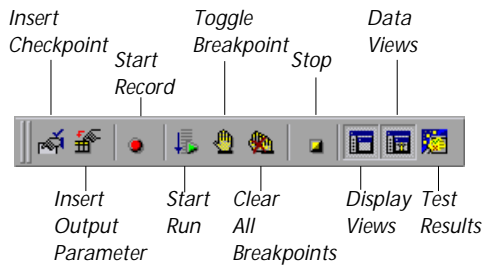
Standard Toolbar

The Standard toolbar contains buttons for managing a test. The following buttons appear on the Standard toolbar:



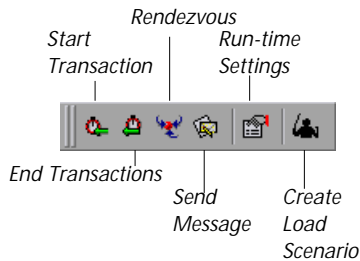
Test Toolbar

The Test toolbar contains buttons for the commands used when creating and maintaining your test. The following buttons appear on the Test toolbar:



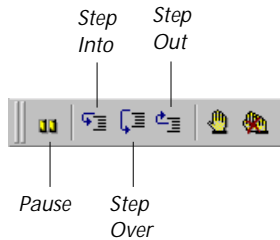
Load Toolbar

The Load toolbar contains buttons for the commands used when creating load testing elements. The following buttons appear on the Load toolbar:



Debug Toolbar

The Debug toolbar contains buttons for the commands used when debugging the steps in your test. The following buttons appear on the Debug toolbar:



Note: The Debug toolbar is automatically enabled when you run your test in the Virtual User Recorder.

Executing Commands Using Shortcut Keys

You can execute some Virtual User Recorder commands by pressing shortcut keys. The following shortcut keys appear on the corresponding menu commands:

Command	Shortcut Key	Function
New	Ctrl + N	Creates a new test and closes your browser.
Open	Ctrl + O	Opens a test.
Save	Ctrl + S	Saves the active test.
Print	Ctrl + P	Prints the active test.
Parameterization/ Properties	Alt + Enter	Opens the Parameterization/Properties dialog box.
Cut	Ctrl + X	Removes the selection from your test.
Copy	Ctrl + C	Copies the selection from your test.
Paste	Ctrl + V	Pastes the selection to your test.
Delete	Del	Deletes the selection from your test.
Rename	F2	Changes the name of an action or a step (Tree View only).
Checkpoint	F12	Creates a checkpoint for a text string, an object, or a table.
Output Parameter	Ctrl + F12	Creates an output parameter for a text string, an object, or a table.
Start	F5	Runs the test from the beginning.
Step Into	F11	Runs only the current line of the test script, however, if the current line calls a test or function, the called test or function is displayed in the view but is not executed.

Command	Shortcut Key	Function
Step Over	F10	Runs only the current line of the test script. When the current line calls another test or a function, the called test or function is executed in its entirety but is not displayed in the view.
Step Out	Shift + F11	Runs to the end of the called test or function, returns to the calling test, and then pauses execution. (Available only after entering a test or function using Step Into.)
Pause	PAUSE	Stops the test run after the statement has been executed. The test run can be resumed from this point.
Toggle Breakpoint	F9	Sets or clears a breakpoint in the test.
Clear All Breakpoints	Ctrl + Shift + F9	Deletes all breakpoints in the test.
Stop	F4	Stops test recording or the test run.

3

Creating Tests

You can quickly create a test script by recording the operations you perform on your Web application.

This chapter describes:

- ▶ Planning a Test
- ▶ Recording a Test
- ▶ Understanding Your Test
- ▶ Managing a Test

About Creating Tests

The Virtual User Recorder enables you to generate an automated test by recording the typical processes that you perform on your Web application. As you navigate through your application, the Virtual User Recorder graphically displays each *step* you perform as an icon in a *test tree*. A step is any action that changes the content of a Web page in your application, for example, clicking a link, or typing data into an edit box.

While recording, you can insert checkpoints into your test. A *checkpoint* searches for a specific value on a Web page and enables you to determine whether or not your Web application is functioning correctly.

When you test your application, you may want to check how it performs the same operations with multiple sets of data. This is called *parameterizing* your test. The data is stored in a *data table*. When you parameterize your test, the Virtual User Recorder substitutes the parameters in your test with

values from the data table. During each iteration of your test, the Virtual User Recorder changes the values in the parameterized statements.

After recording, you can further enhance your test by adding transactions, rendezvous points, and by modifying steps in the test tree.

Planning a Test

Before you start recording, you should plan your test. You should consider the following:

- ▶ Determine the actions you want to record to create your test. Realistic tests that check specific functions and load performance of the application are best.
- ▶ Decide which information you want to check during the test. A checkpoint can check for differences in the text strings, objects, and tables in your application. For more information, see Chapter 4, “Creating Checkpoints.”
- ▶ Consider increasing the power and flexibility of your test by replacing fixed values with parameters. When you parameterize your test, you can check how it performs the same operations with multiple sets of data. For more information, see Chapter 6, “Parameterizing Tests.”

Recording a Test

You create a test by recording the typical processes that users perform on your Web site. The Virtual User Recorder records each step you perform and generates a test tree. Consider the following guidelines when recording a test:

- ▶ Before you start to record, close all applications not required for the test.
- ▶ Determine the security zone of your application. When you record your test, the Web browser may prompt you with security alert dialog boxes. You may choose to disable/enable these dialog boxes.
- ▶ You can control how the Virtual User Recorder records and displays your tests by setting testing options in the Test Settings dialog box. For more information, see Chapter 13, “Setting Testing Options.”

To record a test:



- 1 Open the Virtual User Recorder. For more information, see “Starting the Virtual User Recorder” on page 6.

- 2 Open a test:



➤ To create a new test, click the **New** button or choose **File > New**.



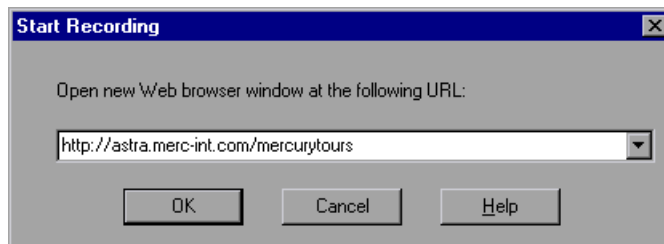
➤ To open an existing test, click the **Open** button or choose **File > Open**. In the **Open Astra Test** dialog box, select a test and click **Open**.

For more information, see “Managing a Test” on page 20.



- 3 Click the **Start Record** button or choose **Record > Start**.



The **Start Recording** dialog box opens.



The URL for Mercury Tours appears as the address. If you would like to open a different browser window, type in the URL for that Web site.

Click **OK**. Your Web browser opens, displaying the Web location you specified.

- 4 Navigate through your application. the Virtual User Recorder records each step you make in the test tree in the **Tree View** tab.
- 5 You can insert text checkpoints, object checkpoints, and table checkpoints to search specific values to determine whether or not an application is functioning correctly. For more information, see Chapter 4, “Creating Checkpoints.”
- 6 You can insert load testing elements to measure how your application functions under load. For more information, see Chapter 5, “Testing Load.”

- 7 You can parameterize your test to check how it performs the same operations with multiple sets of data. For more information, see Chapter 6, “Parameterizing Tests.”
-  8 When you complete your Web session, click the **Stop Record** button or choose **Record > Stop**.
-  9 To save your test, click the **Save** button or choose **File > Save** and assign the test a name. For more information, see “Managing a Test,” on page 20.

Understanding Your Test








While recording, the Virtual User Recorder creates a *test tree*—a graphical representation of the navigations you perform on your application. The test tree appears in the Tree View tab. Each step in the tree represents an action performed on your application and browser.

The following is a sample test of a logon procedure to the Mercury Tours application, Mercury Interactive’s sample Web application.



In the example above, the Web browser invokes the Mercury Tours application. The Virtual User Recorder types “mercury” as the username and password. To complete the logon procedure, the Virtual User Recorder clicks the Login button.

The table below provides an explanation of each step in the tree.

Step	Class	Description
 "Mercury Tours"	browser	The browser invokes the <i>Mercury Tours</i> application.
 "Mercury Tours"	page	The name of the Web page.
 "navbar"	frame	The name of the Web frame.
 Checkpoint "navbar"	checkpoint	A checkpoint that checks the <i>navbar</i> frame.
 "username" Set "mercury"	edit box	<i>username</i> is the name of the edit box. <i>Set</i> is the method performed on the edit box. <i>mercury</i> is the value of the edit box.
 "password" Set "mercury"	edit box	<i>password</i> is the name of the edit box. <i>Set</i> is the method performed on the edit box. <i>mercury</i> is the value of the edit box.
 "Login" Click 776, -95	button	<i>Login</i> is the name of the button. <i>Click</i> is the method performed on the button. <i>776, -95</i> are the x- and y-coordinates where the button was clicked.

Managing a Test

You can create, open, and save recorded tests.

Creating a New Test



To create a new test, click the **New** button or choose **File > New**. You are ready to start recording your test.

Opening an Existing Test

You can open an existing test in order to enhance or run it.

To open an existing test:



- 1 Click the **Open** button or choose **File > Open**. The **Open Astra Test** dialog box opens.
- 2 Select a test and click **Open**. The test opens and the title bar displays the test name.

Saving a Test

You can save a new test or changes to an existing test.

To save a new test:



- 1 Click the **Save** button or choose **File > Save** to save the new test. The **Save Astra Test** dialog box opens.
- 2 Choose the folder in which you want to save the test.
- 3 Type a test name in the **File name** box.
- 4 Click **Save**. Astra LoadTest displays the test name in the title bar.

To save changes to an existing test:



- Click the **Save** button or choose **File > Save** to save changes to the appropriate test.
- Choose **File > Save As** to save an existing test to a new location.

4

Creating Checkpoints

By adding checkpoints to your tests, you can compare pages, text strings, and objects in different versions of your Web application. This enables you to ensure that your Web application functions as desired.

This chapter describes:

- Adding a Page Checkpoint
- Adding a Text or Object Checkpoint
- Checking Text
- Checking Objects
- Modifying Checkpoints

About Creating Checkpoints

The Virtual User Recorder enables you to add checks to your automated test. A *checkpoint* searches for a specific value on a Web page, which enables you to identify whether or not your Web application is functioning correctly. You can check Web pages, text strings, and objects.

When you add a checkpoint to your test, the Virtual User Recorder adds a checkpoint icon under the highlighted step in the test tree. When you run the test, Astra LoadTest compares the expected results of the checkpoint to the current results. If the results do not match, the checkpoint fails. You can view the results of the checkpoint in the Test Results window.

For example, suppose you want to check that a specific text string appears on a page in your Web application. You can add a text checkpoint to check that the text appears in the correct place. You can also check that the text string follows or precedes other text strings.

You can add checkpoints during or after recording a test. It is generally more convenient to define checks after recording—once the initial test has been recorded.

Adding a Page Checkpoint

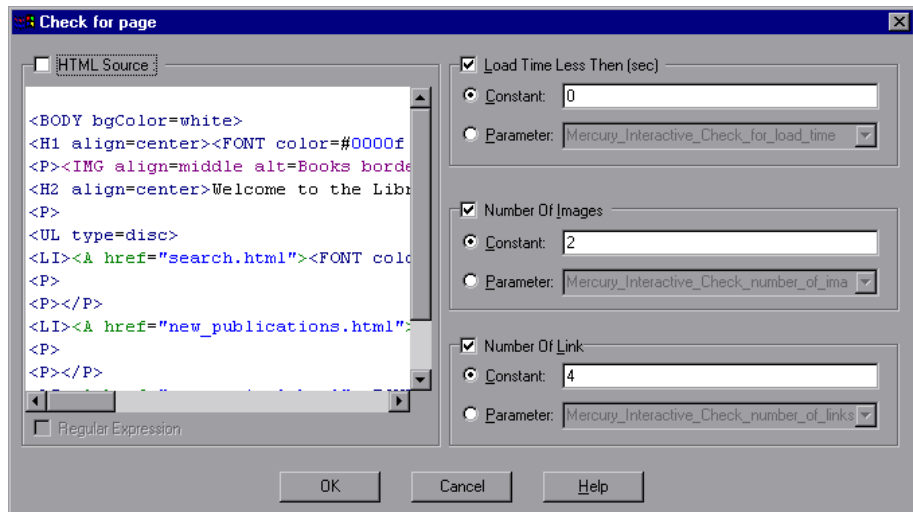
You can check statistical information on your Web page by adding a page checkpoint to your test. To add a page checkpoint to your test, you open the Check for Page dialog box.

In the Check for Page dialog box, you can verify the number of links and images that are displayed on a Web page, check the HTML source code, and verify that the page loads within the time specified.

To add a page checkpoint:



- 1 Make sure the Display Views button is enabled and the ActiveScreen tab is selected.
- 2 Click a step in your test where you want to add a checkpoint. The ActiveScreen displays the Web page corresponding to the highlighted step.
- 3 Right-click the page to check. A menu opens.
- 4 Select Page. Another menu opens.
- 5 Select Statistics. Another menu opens.
- 6 Select Insert Checkpoint. The Check for Page dialog box opens.



7 Specify the settings for the checkpoint in the Check for Page dialog box.

➤ Specifying the load time

In the **Load Time Less Than (sec)** section, you can check that a page loads within the time specified. You can use the following options to specify the load time to check:

Option	Description
Constant (default)	The load time Astra LoadTest checks when running the test.
Parameter	Sets the load time as a parameter. For more information, see Chapter 6, "Parameterizing Tests."

➤ Specifying the number of images

In the **Number of Images** section, you can check the number of images that are displayed on the page. You can use the following options to specify the number of images to check:

Option	Description
Constant (default)	The number of images Astra LoadTest checks when running the test.
Parameter	Sets the number of images as a parameter. For more information, see Chapter 6, "Parameterizing Tests."

➤ **Specifying the number of links**

In the **Number of Links** section, you can check the number of links that are displayed on the page. You can use the following options to specify the number of links to check:

Option	Description
Constant (default)	The number of links Astra LoadTest checks when running the test.
Parameter	Sets the number of links as a parameter. For more information, see Chapter 6, “Parameterizing Tests.”

➤ **Checking the HTML source**

In the **HTML Source** section, you can check the HTML source of the page. You can use the following options to check the HTML source:

Option	Description
HTML Source	Select the HTML Source check box and edit the code. Astra LoadTest checks the code when running the test.
Regular expression	Sets the code as a regular expression. For more information, see Chapter 8, “Using Regular Expressions.”

8 Click **OK** to close the dialog box. A page checkpoint is added to your test.

Adding a Text or Object Checkpoint

In order to create a text or object checkpoint, you must open the relevant dialog box, in which you specify the settings for the checkpoint. You can add a checkpoint while recording or afterward.

To add a text or object checkpoint while recording:

- 1 To check a text string, highlight it. To check an object, proceed to step 2.
- 2 Click the **Insert Checkpoint** button or choose **Insert > Checkpoint**.



The mouse pointer turns into a pointing hand.

- 3 Click the text string or object to check.
- 4 A menu opens.
 - To check a text string in a Web page, select **Text**. The **Text Checkpoint Properties** dialog box opens.
 - To check an object, select the first menu option. The option name depends on your object, for example:

Object	Menu Option
Check box	WebCheckBox
Edit box	WebEdit
Image	Image
Radio button	WebRadio

The **Object Checkpoint** dialog box opens.

- 5 Specify the settings for the checkpoint in the open dialog box.
 - For information on the Text Checkpoint Properties dialog box, see “Checking Text,” on page 29.
 - For information on the Object Checkpoint dialog box, see “Checking Objects,” on page 32.
- 6 Click OK to close the dialog box.

To add a text or object checkpoint after recording:



- 1 Make sure the Display Views button is enabled and the ActiveScreen tab is selected.
- 2 Click a step in your test where you want to add a checkpoint. The ActiveScreen displays the Web page corresponding to the highlighted step.
- 3 Click the text string or object to check.
- 4 A menu opens.
 - To check a text string in a Web page, select Text. Another menu opens. Select Insert Checkpoint. The Text Checkpoint Properties dialog box opens.
 - To check an object, select the first menu option. The option name depends on your object, for example:

Object	Option Name
Check box	WebCheckBox
Edit box	WebEdit
Image	Image
Radio button	WebRadio

Another menu opens. Select Insert Checkpoint. The Object Checkpoint dialog box opens.

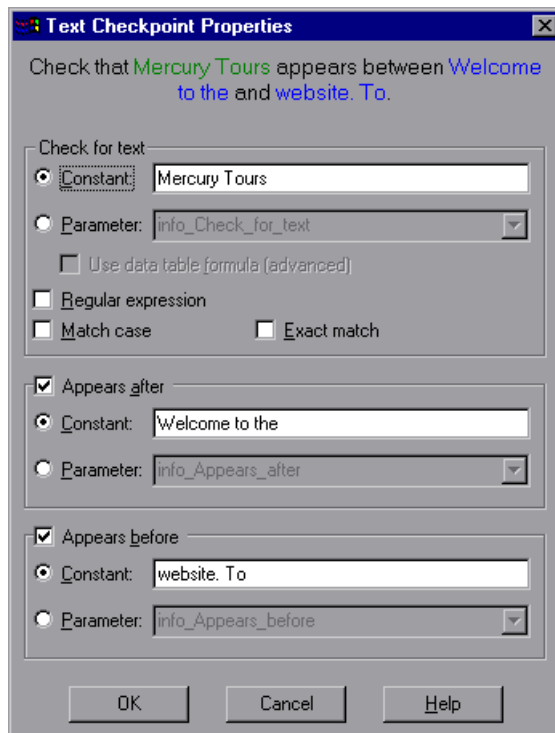
- 5 Specify the settings for the checkpoint in the open dialog box.

- For information on the Text Checkpoint Properties dialog box, see “Checking Text,” on page 29.
 - For information on the Object Checkpoint dialog box, see “Checking Objects,” on page 32.
- 6 Click OK to close the dialog box. A checkpoint is added to your test.

Checking Text

You can check that a specified string appears on your Web page by adding a text checkpoint to your test. To add a text checkpoint to your test, you open the Text Checkpoint Properties dialog box, as described in “Adding a Text or Object Checkpoint,” on page 26.

In the Text Checkpoint Properties dialog box, you can specify which text to check as well as which text appears before and after the text to check. This is particularly helpful when the text string you want to check appears several times in the same Web page. For example, suppose you want to check that the “Mercury Tours” text string appears in a specific location in the first page of the sample Web flight application, “Mercury Tours. This text string actually appears three times on that Web page. To check for the text string in a specific location, you can specify which text precedes or follows the text string you are checking.



Specifying which Text to Check

In the **Check for text** section, you use the following options to specify which text to check:

Option	Description
Constant (default)	The text the Virtual User Recorder checks when running the test.
Parameter	Sets the text string as a parameter. For more information, see Chapter 6, "Parameterizing Tests."
Use data table formula (advanced)	Adds a parameter and an output parameter. For more information, see Chapter 6, "Parameterizing Tests," and Chapter 7, "Creating Output Parameters."
Regular expression	Sets the text string as a regular expression. For more information, see Chapter 8, "Using Regular Expressions."
Match case	Conducts a case sensitive search.
Exact match	Checks according to the exact expected text.

Specifying What Appears After the Text to Check

In the **Appears after** section, you use the following options to specify which text, if any, should appear after the text to check:

Option	Description
Appears after (default)	Checks that the information in the section appears after the text to check. To ignore the text that appears after the text to check, clear this check box.
Constant (default)	Displays the text that appears after the text to check.
Parameter	Sets the text string as a parameter. For more information, see Chapter 6, "Parameterizing Tests."

Specifying What Appears Before the Text to Check

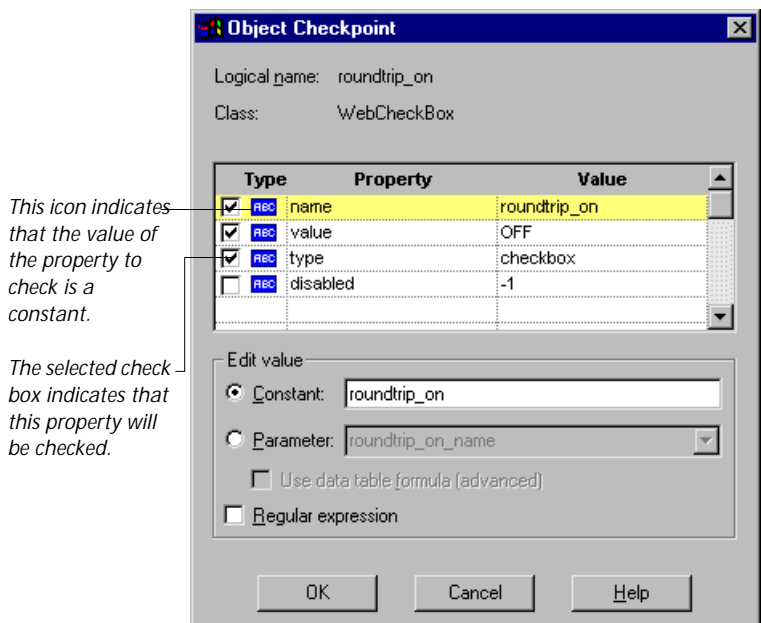
In the **Appears before** section, you use the following options to specify which text, if any, should appear before the text to check:

Option	Description
Appears before (default)	Checks that the information in the section appears before the text to check. To ignore the text that appears before the text to check, clear this check box.
Constant (default)	Displays the text that appears before the text to check.
Parameter	Sets the text string as a parameter. For more information, see Chapter 6, “Parameterizing Tests.”

Checking Objects

You can check that a specified object appears on your Web page by adding an object checkpoint to your test. To add an object checkpoint to your test, you open the Object Checkpoint dialog box, as described in “Adding a Text or Object Checkpoint,” on page 26.

In the Object Checkpoint dialog box, you can specify which properties of the object to check, and edit the values of these properties.





Identifying the Object

The top part of the dialog box displays information about the object to check:

Information	Description
Logical name	The name of the object as defined in the HTML code of the Web page.
Class	The type of object. In this example, the "WebCheckBox" class indicates that the object is a check box.

Choosing which Property to Check

The dialog box also displays the properties of the object you can check in the Properties pane, which lists the properties, their values, and their types:

Option	Description
check box	<p>For each object class, the Virtual User Recorder recommends default property checks. You can accept the default checks or modify them accordingly.</p> <p>To check a property, select the corresponding check box.</p> <p>To exclude a property check, clear the corresponding check box.</p>
Type	<p>The  icon indicates that the value of the property is a constant.</p> <p>The  icon indicates that the value of the property is a parameter.</p>
Property	The name of the property to check.
Value	<p>The value of the property to check. Note that unless you edit this value, it will be the expected value of the property when you run your test. For information about editing the value of a property, see “Editing the Value of an Object’s Property,” on page 34.</p>

Editing the Value of an Object's Property

In the **Edit value** section, you use the following options to edit the value of the property to check.

Option	Description
Constant (default)	Sets the value of the property.
Parameter	Sets the property value as a parameter. For more information, see Chapter 6, "Parameterizing Tests."
Use data table formula (advanced)	Adds a parameter and an output parameter. For more information, see Chapter 6, "Parameterizing Tests," and Chapter 7, "Creating Output Parameters."
Regular expression	Sets the property value as a regular expression. For more information, see Chapter 8, "Using Regular Expressions."

Modifying Checkpoints

You can modify checkpoints in your test.

To modify a checkpoint:

- 1 Right-click a checkpoint in your test. A menu opens.
- 2 Select **Parameterization/Properties**. A checkpoint dialog box opens.
- 3 Modify the properties and click **OK**.

5

Testing Load

Today's Web-based applications are accessed by multiple application clients over complex architectures. With Astra LoadTest, you can emulate the load of real users interacting with your application and measure system performance.

The Virtual User Recorder enables you to customize your test to accurately measure the performance of your Web application under load.

This chapter describes:

- Inserting Transactions
- Inserting Rendezvous points
- Setting Run-time Options
- Run-time Settings
- Sending Messages to Output

About Transactions

To measure the performance of the server, you define *transactions*. A transaction represents an action or a set of actions that you are interested in measuring. You define transactions within your Vuser script by enclosing the appropriate sections of the script with *start* and *end* transaction statements. For example, you can define a transaction that measures the time it takes for the server to process a request to reserve a seat on a flight and for the confirmation to be displayed at the application client's terminal.

When you record a test, Astra LoadTest automatically marks each page you browse as a transaction. This means that when you run a scenario, each page in your test tree is recognized as a transaction to be measured.

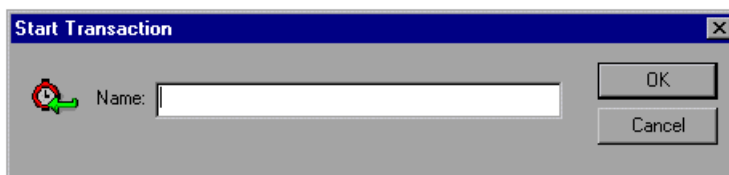
The automatic transactions create a great deal of general analysis information. As you refine your test, you may want to remove the automatic transactions and insert transactions that will measure the performance of specific business processes. You manually insert a transaction to mark a group of steps that make up the business process that you want to measure.

To insert a transaction:

- 1 In the test tree, click the step where you want your transaction timing to begin. The page opens in the ActiveScreen.



- 2 Click the Start Transaction button. The Start Transaction dialog box opens.



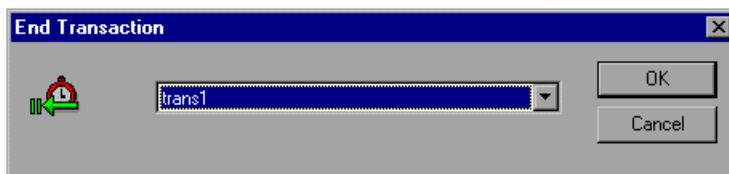
- 3 Enter a meaningful name in the Name box. Click OK.

The Start Transaction icon is added to the test tree below the highlighted step.

- 4 In the test tree, click the step where you want the transaction timing to end. The page opens in the ActiveScreen.



- 5 Click the End Transaction button. The End Transaction dialog box opens.



- 6 The Name box contains the transaction name you entered in the Start Transaction dialog box. Click OK.

The End Transaction icon is added to the test tree above the selected step.

Note: There is no limit to the number of Transactions you can add to a test.

About Rendezvous Points

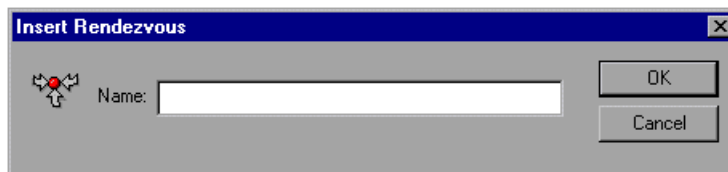
During the scenario run, you instruct multiple Vusers to perform tasks simultaneously by creating a rendezvous point. This ensures that:

- intense user load is emulated
- transactions are measured under the load of multiple Vusers

A rendezvous point is a meeting place for Vusers. To designate the meeting place, you insert rendezvous statements into your Vuser scripts. When the rendezvous statement is interpreted, the Vuser is held by the Controller until all the members of the rendezvous arrive. When all the Vusers have arrived (or a time limit is reached), they are released together and perform the next task in their Vuser scripts.

To insert a rendezvous point

- 1 In the test tree, click the step where you want your intense user load emulated. The page opens in the ActiveScreen.
- 2 Click the Rendezvous button. The Rendezvous dialog box opens.



- 3 Enter a meaningful name in the Name box. Click **OK**.

About Setting Run-time Options

Astra LoadTest run-time options affect how your test runs in a load testing scenario. For example, you can set the number of iterations which Astra LoadTest will run, or set the output messages sent by Astra LoadTest.

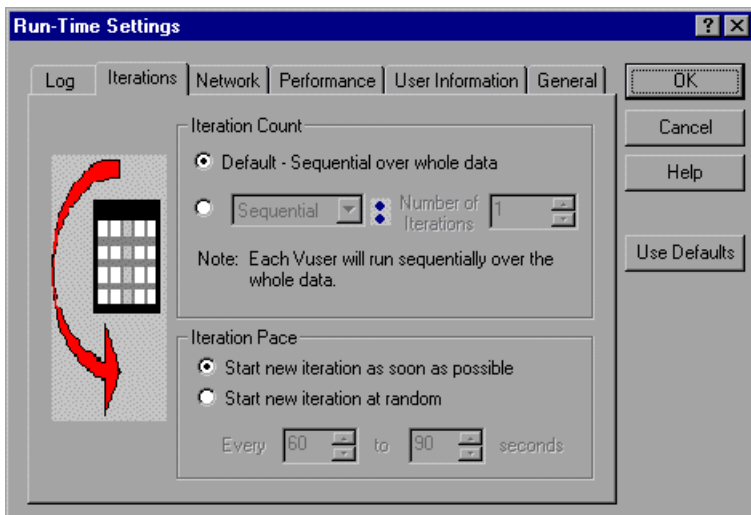
Setting Run-time Options

Before you run your test, you can use the Run-time Settings dialog box to modify your testing options. The values you set remain in effect for all tests.

To set run-time options:

- 1 Choose Load > Run-Time Settings.

The Run-Time Settings dialog box opens. It is divided by subject into six tabbed pages.



- 2 Set an option, as described in “Run-time Settings,” on page 39.
- 3 When you are done, click OK to apply your changes and close the dialog box.

Run-time Settings

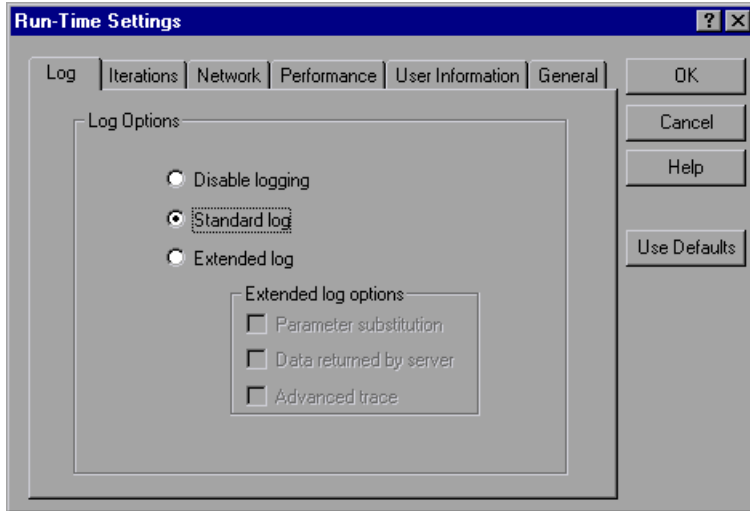
The Run-Time Settings dialog box contains the following tabbed pages:

Tab Heading	Subject
Log	Options for output messages
Iterations	Options for the number of iterations of a test and their pacing
Network	Options on how to handle Web images, Java applets and scripts
Performance	Options for think time
User Information	Options to handle network passwords
General	Options for run mode and global settings

This section lists the testing options you can set using the Run-Time Settings dialog box.

Log Settings

The Log tab options indicate what type of output messages Astra LoadTest should send to the output file.



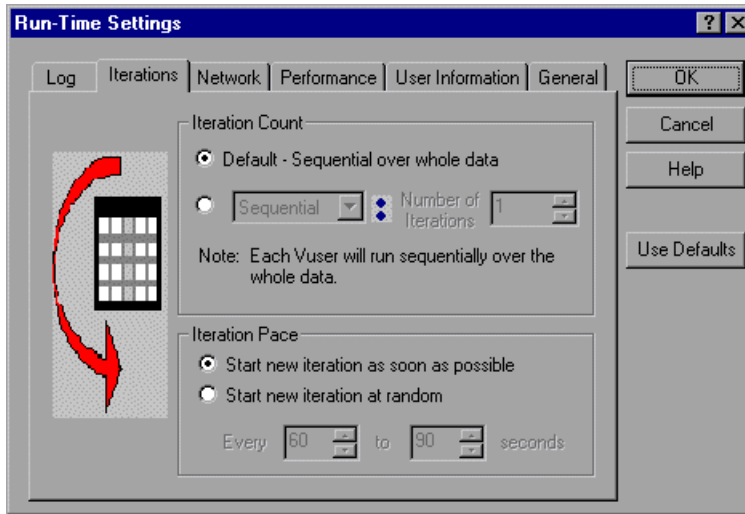
The Log tab includes the following options:

Option	Description
Disable logging	Instructs Astra LoadTest not to send output messages.
Standard log	Instructs Astra LoadTest to send standard output messages. This is the default setting.
Extended log	Instructs Astra LoadTest to send detailed output messages to the output file.
Parameter substitution	Instructs Astra LoadTest to log all the parameters that are replaced while a script runs, and the values that replace the parameters.
Data returned by server	Instructs Astra LoadTest to log all the data that is returned by the server.
Advanced trace	Instructs Astra LoadTest to log all functions called and messages sent by the Vuser during the session.

Note: The use of the log option results in a higher usage of resources, therefore, disable the log options to improve scalability

Iterations Settings

The Run-Time Settings dialog box:Log tab;Log tab, Run-Time Settings dialog box;setting:Log options in the Run-Time Settings dialog boxIterations tab options determine how many times to run a test, and how often.

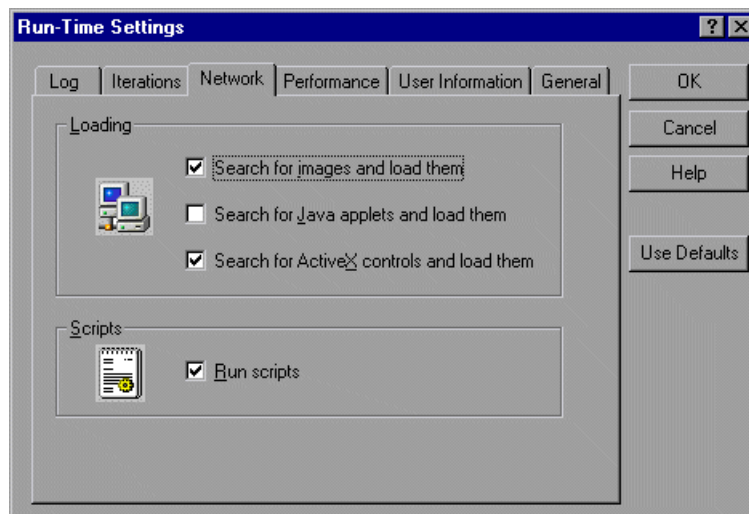


The Iterations tab includes the following options:

Option	Description
Default - Sequential over whole data	Each Vuser will run as many times as it takes to run through the data sets sequentially.
Sequential	The data is taken from the data table according to the sequential order of the rows.
Number of Iterations	Sets the number of iterations to run.
Start new iteration as soon as possible	Instructs Astra LoadTest to start a new iteration as soon as the previous one is complete.
Start new iteration at random	Starts a new iteration after a random interval.
Every _ to _ seconds	Specifies the minimum and the maximum time of the random interval - in seconds.

Network Settings

The Network tab specifies how the Vuser handles Web images, Java applets and scripts.

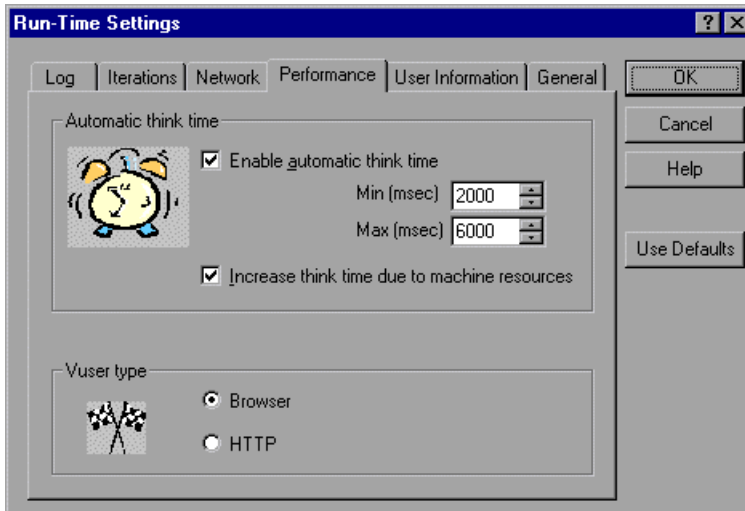


The Network tab includes the following options:

Option	Description
Search for images and load them	Instructs a Vuser to load the graphics associated with a Web page when the Vuser accesses the page during script execution.
Search for Java applets and load them	Instructs a Vuser to load the Java applets associated with a Web page when the Vuser accesses the page during script execution.
Search for ActiveX controls and load them	Instructs a Vuser to load the ActiveX controls associated with a Web page when the Vuser accesses the page during script execution.
Run Scripts	Instructs a Vuser to run scripts associated with a Web page when the Vuser accesses the page during script execution.

Performance Settings

The Performance tab gives you the option to set *think time*. Think time emulates the time that a real user waits between actions.

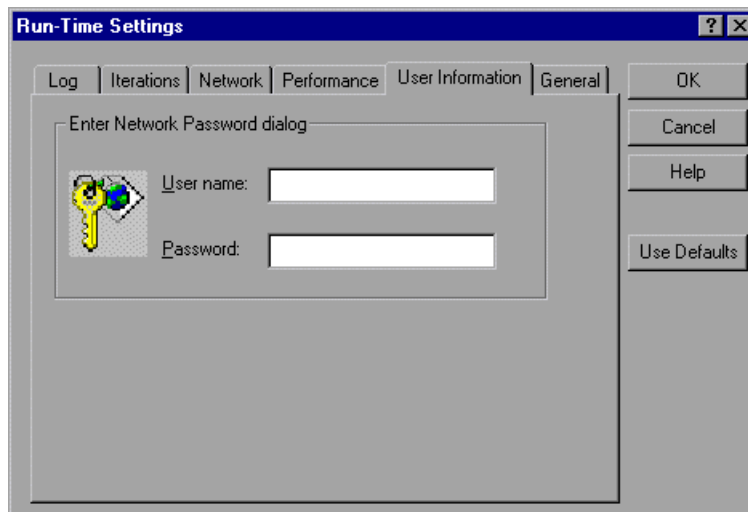


The Performance tab includes the following options:

Option	Description
Enable automatic think time	Allows you to designate a range, in milliseconds, for the Vuser to wait between steps. Within the range the think times are random.
Min(msec)	Sets the minimum wait time in milliseconds.
Max(msec)	Sets the maximum wait time in milliseconds.
Increase think time due to machine resources	Increases the think time to compensate for slower processing times due to hardware limitations.
Browser	Sets the Browser as the replay mode. This is the default setting and is used for standard testing .
HTTP	Sets HTTP as the replay mode. This setting is used when there is a need for increased scalability and performance when running a test.

User Information Settings

The User Information tab allows you to enter network passwords for processing tests in applications where passwords are required.

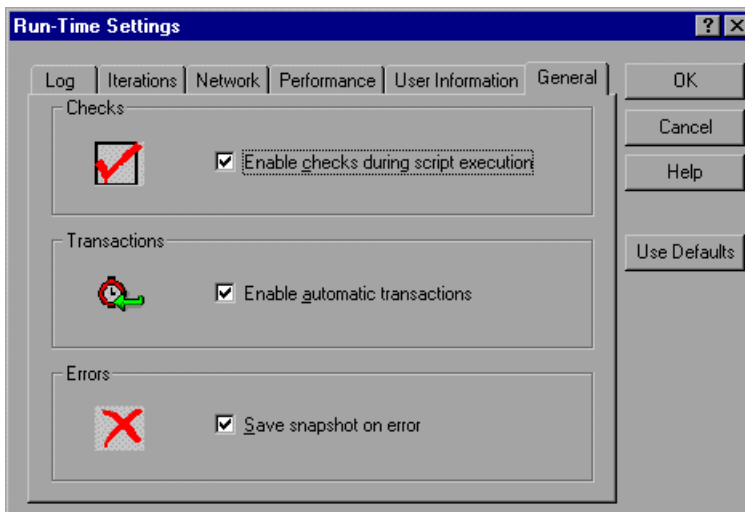


The User Information tab includes the following options:

Option	Description
User	The Network user name.
Password	The Network password.

General Settings

The General tab enables Astra LoadTest to perform checks or transactions during a test run.



The General tab includes the following options:

Option	Description
Enable checks during script execution	Instructs Astra LoadTest to perform checks during script execution.
Enable automatic transactions	Instructs Astra LoadTest to measure each browsed page as a transaction.
Save snapshot on error	Instructs Astra LoadTest to save an image of a Web page where an error is detected.

Sending Messages to Output

When you run a scenario, the Controller's Output window displays information about test execution. You can include statements in a Vuser test to send error and notification messages to the Controller. The Controller displays these messages in the Output window. For example, you could insert a message that displays the current state of the Web application. You can also save these messages to a file.

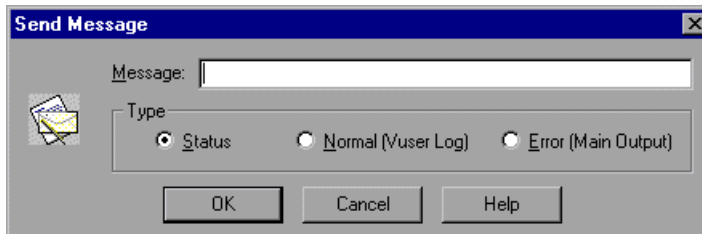
Note: Do not send messages from within a transaction. Doing so lengthens the transaction execution time and may skew the actual transaction results.

To send messages to the output window:

- 1 Click a step in your test where you want to send a message.



- 2 Click the **Send Message** button or choose **Load > Send Message**. The **Send Message** dialog box opens.



- 3 Type the message into the **Message** box.

- 4 Specify the message type.

The message type includes the following options:

Option	Description
Status	Instructs Astra LoadTest to generate and print formatted output to the Controller Vuser status area.
Normal(Vuser Log)	Instructs Astra LoadTest to send the output message directly to a file.
Error(Main Output)	Instructs Astra LoadTest to send an error message to the Output window.

- 5 Click **OK** to close the dialog box. The **Send Message** icon is added above the highlighted step.

6

Parameterizing Tests

Astra LoadTest enables you to expand the scope of a basic test by replacing fixed values with parameters. This process, known as *parameterization*, greatly increases the power and flexibility of your tests.

This chapter describes:

- ▶ Parameterizing Steps
- ▶ Parameterizing Checkpoints
- ▶ Editing the Data Table

About Parameterizing Tests

You can use the Virtual User Recorder to enhance your tests by parameterizing values in the test. A *parameter* is a variable that is assigned a value from outside the test in which it is defined.

You start by recording a test that performs a set of actions. After you finish recording, you can parameterize certain constants in the test so that the test will run the same set of actions many times. In each test run, Astra LoadTest substitutes a constant for the value of the parameter. You supply the list of possible values for a parameter in the data table in the test.

	Departure	Arrival	C	D	E
1	New York	Paris			
2	London	New York			
3	Paris	Frankfurt			
4					
5					
6					

Each *column* in the table represents the list of values for a single parameter. The column header is the parameter name.

Each *row* in the data table represents a set of values that Astra LoadTest submits for all the parameters during a single iteration of the test. When you run your test, Astra LoadTest runs one iteration of the test for each set of data in the data table. For example, a test with a ten-row data table will run ten times.

For example, consider the sample Web flight application, “Mercury Tours,” which enables you to book flight requests. To book a flight, you supply the flight itinerary and click the Purchase Flight button. The application returns the available flights for the requested itinerary.

You could conduct the test by accessing the Web site and recording the submission of numerous queries. This is a slow, laborious, and inefficient solution. When you parameterize your test, you first record a test that accesses the Web site and checks for the available flights for one requested itinerary. You then substitute the recorded itinerary for a variable, and add multiple sets of data, one for each itinerary, into the data table linked to the test. When you run the test, Astra LoadTest submits a separate query for each itinerary.

When you add parameters to your test, you can parameterize a step recorded in your test or a checkpoint added to your test. When you parameterize a step, you parameterize either the *object* that you navigate in your Web page or the *method* by which you navigate. When you parameterize a checkpoint, instead of checking how your Web application performs an operation on a

single text string or object, you can check how it performs with multiple sets.

You can also parameterize your test by creating output parameters, which retrieve variables from the test while it runs and insert them into the test's data table. For additional information, see Chapter 7, "Creating Output Parameters."

Parameterizing Steps

You can parameterize a step while recording your test or afterward. You parameterize a step in your test tree. A step is made up of an *object* that you navigate in your Web page, and/or a *method* by which you navigate the step. When you parameterize a step, you are actually parameterizing either the object or the method.

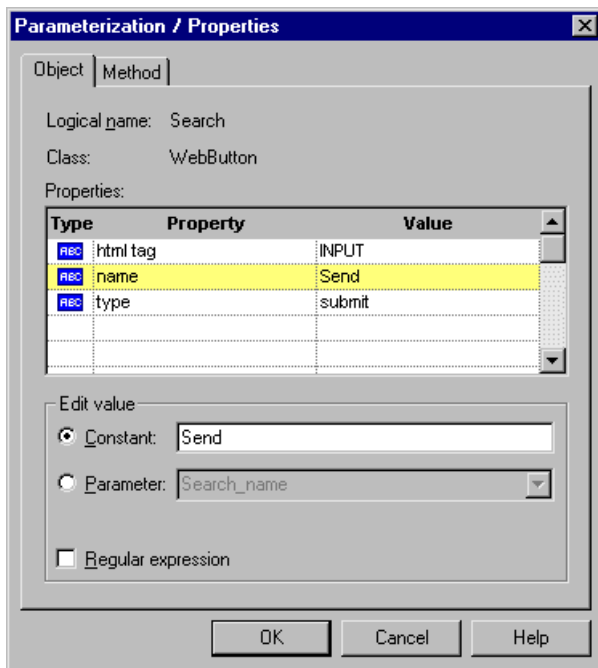
Parameterizing an Object in a Step

You can parameterize the object that you navigate in a step. For example, your Web application may include a form in which the user can click one of many radio buttons. You may want to test how your application responds when different radio buttons are selected. Rather than record a separate test for clicking each radio button, you can parameterize your test so that during each iteration of the test run, Astra LoadTest clicks a different radio button.

To parameterize an object in a step:

- 1 In the test tree, right-click a step. A menu opens.
- 2 Select **Parameterization/Properties**. The **Parameterization/Properties** dialog box opens and displays the properties of the object and/or the arguments of the method in the step.



3 To parameterize the object in a step, make sure the Object tab is selected.




The Object tab displays information about the object in the step:

Information	Description
Logical name	The name of the object as defined in the HTML code of the Web page.
Class	The type of object. In this example, the "WebButton" class indicates that the object is a button.

The dialog box displays the properties you can parameterize, in a pane listing the properties, their values, and their types:

Option	Description
Type	The  icon indicates that the property value is a constant. The  icon indicates that the property value is a parameter.
Property	The name of the property whose value will be parameterized.
Value	The value of the property to parameterize.

- 4 Click the property to parameterize in the **Properties** section. The property is highlighted.
- 5 In the **Edit value** section, click **Parameter**.
- 6 In the **Parameter** box, choose a parameter from the list or enter a new name.
 - To use a parameter that you already created, select it from the list.
 - To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter.
- 7 If you want to set the property value of the step as a regular expression, select the **Regular expression** check box. For more information, see Chapter 8, “Using Regular Expressions.”
- 8 Click **Close** to save the parameter and close the dialog box.
- 9 If you created a new parameter, the **Astra parameters** dialog box prompts you to add the new parameter to the data table. Click **OK**. A new column is highlighted in the table for the new parameter.

In your test tree, the  icon next to the step indicates that the step has been parameterized.

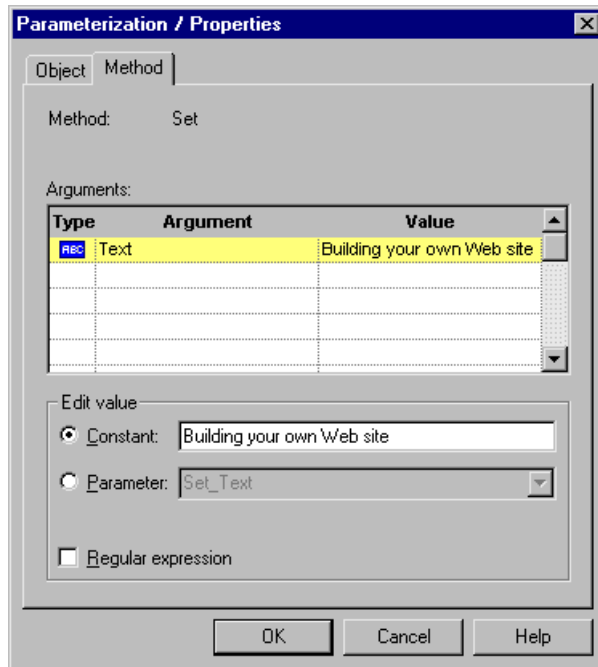
Note: You can specify additional data values for the parameter by entering them directly into the data table. For more information, see “Editing the Data Table” on page 58

Parameterizing a Method in a Step

You can parameterize the method you use to navigate a step. For example, your Web application may include a form with an edit field into which the user types a text string. You may want to test how your application responds to different data in the form. Rather than record a separate test for each text string typed, you can parameterize your test so that during each iteration of the test run, Astra LoadTest enters a different text string into the edit field.

To parameterize a method in a step:



- 1 In the test tree, right-click a step. A menu opens.
- 2 Select Parameterization/Properties. The Parameterization/Properties dialog box opens and displays the properties of the object and/or the arguments of the method in the step.
- 3 To parameterize the method in a step, make sure the Method tab is selected.




The Method tab displays the name of the method in the step:

Information	Description
Method	The name of the method as defined in the HTML code of the Web page.

The dialog box displays the arguments you can parameterize, in a pane listing the arguments, their values, and their types:

Option	Description
Type	The  icon indicates that the argument value is a constant. The  icon indicates that the argument value is a parameter.
Argument	The name of the argument whose value will be parameterized.
Value	The value of the argument to parameterize.

- 4 Click an argument in the **Arguments** section. The argument is highlighted.
- 5 In the **Edit value** section, click **Parameter**.
- 6 In the **Parameter** box, choose a parameter from the list or enter a new name.
 - To use a parameter that you already created, select it from the list.
 - To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter.
- 7 If you want to set the argument value of the step as a regular expression, select the **Regular expression** check box. For more information, see Chapter 8, “Using Regular Expressions.”
- 8 Click **Close** to save the parameter and close the dialog box.
- 9 If you created a new parameter, the **Astra parameters** dialog box prompts you to add the new parameter to the data table. Click **OK**. A new column is highlighted in the table for the new parameter.

In your test tree, the  icon next to the step indicates that the step has been parameterized.

Note: You can specify additional data values for the parameter by entering them directly into the data table. For more information, see “Editing the Data Table” on page 58.


Parameterizing Checkpoints

You can parameterize a checkpoint while recording your test or afterward. For information on parameterizing checkpoints while creating them, see Chapter 4, “Creating Checkpoints.”

When you test your Web site, you may want to check how it performs the same operations with multiple sets of data. For example, if you are testing the sample flight application, “Mercury Tours,” you may create a checkpoint to check that once you book a ticket, it is booked correctly. Suppose that you want to check that flights are booked correctly for a variety of different destinations. Rather than create a separate test with a separate checkpoint for each destination, you can parameterize the destination information: for each iteration of the test, Astra LoadTest checks the flight information for a different destination.

To parameterize a checkpoint after recording your test:

- 1 In the test tree, right-click a checkpoint. A menu opens.
- 2 Select Parameterization/Properties.
 - For a text checkpoint, the Text Checkpoint Properties dialog box opens.
 - For an object checkpoint, the Object Checkpoint dialog box opens.
- 3 In the open dialog box, click Parameter to set the value as a parameter.
- 4 In the Parameter box, choose a parameter from the list or enter a new name.
 - To use a parameter that you already created, select it from the list.
 - To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter.
- 5 Click OK to save the parameter and close the dialog box.
- 6 If you created a new parameter, the Astra Parameter dialog box prompts you to add the new parameter to the data. Click OK. A new column is highlighted in the table for the new parameter.

In your test tree, the  icon next to the checkpoint indicates that the checkpoint has been parameterized.


Note: You can specify additional data values for the parameter by entering them directly into the data table. For more information, see “Editing the Data Table” on page 58.

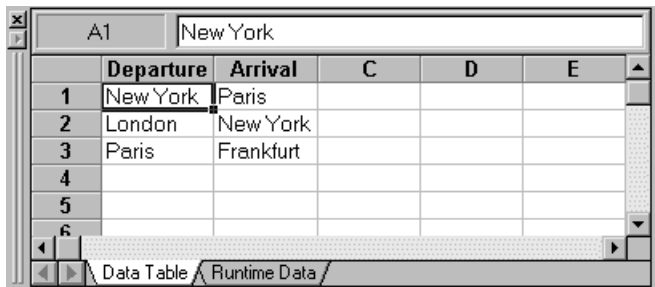
Editing the Data Table

The data table contains the values that Astra LoadTest substitutes for parameters when you run a test. Astra LoadTest automatically saves the data table for a test in the test folder and assigns it a *.xls* extension. Whenever you save your test, Astra LoadTest automatically saves the test's data table.

You can edit information in the data table by typing directly into the table. You can also import data in Excel 95, Excel 97, or ASCII format. You use the data table in the same way as an Excel spreadsheet, including inserting formulas into cells.

To edit the data table:

- 1 Open your test.
- 2  Make sure the Data Views button is enabled and the Data Table tab is selected.



	Departure	Arrival	C	D	E
1	New York	Paris			
2	London	New York			
3	Paris	Frankfurt			
4					
5					
6					

- Each *row* in the data table represents the values that Astra LoadTest submits for all the parameterized arguments during a single iteration of the test. For example, a test that is associated with a table with ten rows will run ten times.
- Each *column* in the table represents the list of values for a single parameterized argument. The column header is the parameter name.

Note: You must enter data in rows from top to bottom, i.e., you cannot enter data in a cell in a row until you have entered data in a cell in a previous row.

- 3 To change the name of a column, double-click on the column heading cell. The **Change Parameter Name** dialog box opens. Type a parameter name and click **OK**.
- 4 Use the menu commands described below to edit the data table. To open the menu, right-click a cell. The following menus are available: **File**, **Edit**, **Data**, and **Format**.

File Menu

The following commands are available in the **File** menu:

File Command	Description
Export	Saves the data table as a file.
Print	Prints the data table.

Edit Menu

The following commands are available in the Edit menu:

Edit Command	Description
Copy	Copies the data table selection and puts it on the Clipboard.
Paste	Pastes the contents of the Clipboard to the current data table selection.
Paste Values	Pastes values from the Clipboard to the current data table selection. Any formatting applied to the values is ignored. In addition, only formula results are pasted; formulas are ignored.
Clear	Clears formats or contents from the current selection. You can clear only formats, only contents (including formulas), or both formats and contents.
Insert	Inserts empty cells at the location of the current selection. Cells adjacent to the insertion are shifted to make room for the new cells.
Delete	Deletes the current selection. Cells adjacent to the deleted cells are shifted to fill the space left by the vacated cells.
	Finds a cell containing specified text and replaces it with different text. You can search by row or column in the table and specify to match case or find entire cells only. You can also replace all.

Data Menu

The following commands are available in the Data menu:

Data Command	Description
Sort	Sorts a selection of cells by row or column and keys.
AutoFill List	Creates, edits, or deletes an autofill list. An autofill list contains frequently-used series of text such as months and days of the week. When adding a new list, separate each item with a semi-colon. To use an autofill list, enter the first item into a cell in the data table. Drag the cursor across or down and the Virtual User Recorder automatically fills in the cells in the range according to the autofill list.

Format Menu

The following commands are available in the Format menu:

Format Command	Description
Currency(0)	Sets format to currency with commas and no decimal places.
Currency(2)	Sets format to currency with commas and two decimal places.
Fixed	Sets format to fixed precision with commas and no decimal places.
Percent	Sets format to percent with no decimal places. Numbers are displayed as percentages with a trailing percent sign (%).
Fraction	Sets format to fraction.
Scientific	Sets format to scientific notation with two decimal places.
	Sets validation rule to test data entered into a cell or range of cells. A validation rule consists of a formula to test, and text to display if the validation fails.

7

Creating Output Parameters

Astra LoadTest enables you to parameterize your test by retrieving a variable value from your test and entering it in your data table as an output parameter. You can subsequently use this output parameter as an input variable in your test. This enables you to use data retrieved during a test in other parts of the test.

This chapter describes:

- Creating Output Parameters
- Parameterizing Text Output
- Parameterizing Object Output
- Parameterizing Table Output

About Creating Output Parameters

You parameterize your test by adding values to a data table which replace variables in the test. When you run the test, Astra LoadTest runs one iteration of the test for each set of values from your data table, as discussed in Chapter 6, “Parameterizing Tests.”

You can also parameterize your test by creating output parameters. An *output parameter* is a value retrieved from a parameter in your test while the test runs and entered into your data table.

For example, consider a flight reservation application. You design a test to create a new reservation and then view the reservation details. Every time you run the test, the application generates a unique order number for the new reservation. To view the reservation, the application requires the user to

input the same order number. However, you cannot know the order number before you run the test.


To solve this problem, you create an output parameter for the unique order number that the application generates when creating a new reservation. In the view reservation screen, you parameterize the order number input field. You use the same output parameter as the unique order number.

When you run the test, Astra LoadTest retrieves the unique order number generated by the application for the new reservation and inserts it in the **Runtime Data** tab of the data table for the order number output parameter. When the test reaches the order number input field required to view the reservation, Astra LoadTest uses the unique order number stored in the data table for the order number input field parameter.

Creating Output Parameters

You can create an output parameter while recording your test or afterward. When you create an output parameter, you parameterize a constant text string or object by replacing it with a variable. For information on creating an output parameter while creating a checkpoint, see Chapter 4, “Creating Checkpoints.”

To create an output parameter while recording:

- 1 To parameterize a text string, highlight it. To parameterize an object, proceed to step 2.
- 2  Click the **Insert Output Parameter** button or choose **Insert > Output Parameter**.

The mouse pointer turns into a pointing hand.

- 3 Click the text string or object to parameterize.

4 A menu opens.

- To parameterize a text string in a Web page, select **Text**. The **Text Output Parameter Properties** dialog box opens.
- To parameterize an object, select the first option. The option name depends on your object, for example:

Object	Option Name
Check box	WebCheckBox
Edit box	WebEdit
Image	Image
Radio button	WebRadio

The **Object Output Parameter** dialog box opens.

- To parameterize a text string in a table, select **WebTable**. The **Table Output Parameter Properties** dialog box opens.

5 Specify the settings for the output parameter in the open dialog box.

- For information on the **Text Output Parameter Properties** dialog box, see “Parameterizing Text Output” on page 68.
- For information on the **Object Output Parameter** dialog box, see “Parameterizing Object Output” on page 70.
- For information on the **Table Output Parameter Properties** dialog box, see “Parameterizing Table Output” on page 72.

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

To create an output parameter after recording:

1 Make sure the **Display Views** button is enabled and the **ActiveScreen** tab is selected.

2 Click a step in your test where you want to create an output parameter.

The **ActiveScreen** displays the Web page corresponding to a highlighted step.

3 Right-click the text string or the object to parameterize.

4 A menu opens.

- To parameterize a text string in a Web page, select **Text**. Another menu opens. Select **Output**. The **Text Output Parameter Properties** dialog box opens.
- To parameterize an object, select the first option. The option name depends on your object, for example:

Object	Option Name
Check box	WebCheckBox
Edit box	WebEdit
Image	Image
Radio button	WebRadio

Another menu opens. Select **Output**. The **Object Output Parameter** dialog box opens.

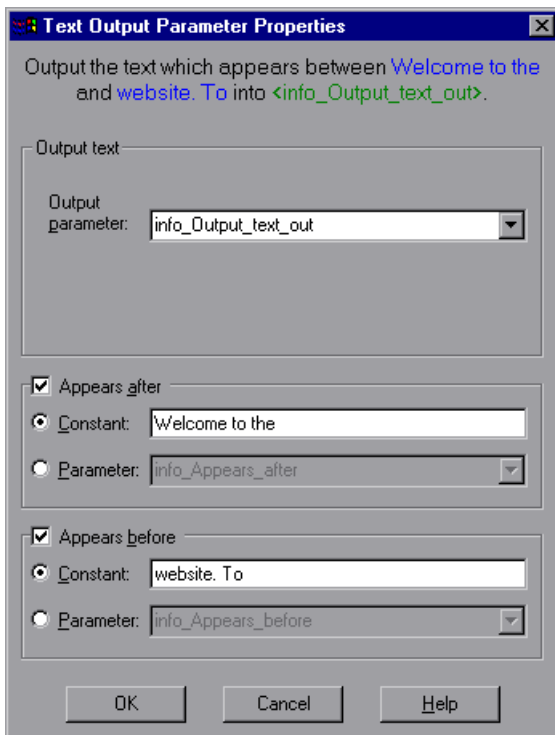
- To parameterize a text string in a table, select **WebTable**. Another menu opens. Select **Output**. The **Table Output Parameter Properties** dialog box opens.

- 5 Specify the settings for the output parameter in the open dialog box.
 - For information on the Text Output Parameter Properties dialog box, see “Parameterizing Text Output” on page 68.
 - For information on the Object Output Parameter dialog box, see “Parameterizing Object Output” on page 70.
 - For information on the Table Output Parameter Properties dialog box, see “Parameterizing Table Output” on page 72.
- 6 Click OK to close the dialog box.
- 7 If you created a new output parameter, the Astra parameters dialog box prompts you to add the new output parameter to the data table. Click OK. A new column is highlighted in the data table for the new output parameter. An output statement is added to your test tree.

Parameterizing Text Output

You can parameterize a text string on your Web page to create a text output parameter. To parameterize a text string, you open the Text Output Parameter Properties dialog box, as described in “Creating Output Parameters” on page 64.

In the Text Output Parameter Properties dialog box, you can specify which text to parameterize as well as which text appears before and after the parameter. This is particularly helpful when the text string you want to parameterize appears several times in the same Web page. For example, suppose you want to parameterize the “Mercury Tours” text string in a specific location in the first page of the sample Mercury Tours Web site. This text string actually appears three times on that Web page. To parameterize the text string in a specific location, you can specify which text precedes or follows the text string you are parameterizing.



Specifying which Text to Parameterize

In the **Output text** section, the Output parameter box displays the default output parameter name for the highlighted text string. In the **Output Parameter** box, you can create a new output parameter or an existing output parameter.

- To use an output parameter that you already created, select an output parameter from the list.
- To create a new output parameter, you can use the default output parameter name, or type a descriptive name for the output parameter.

Specifying What Appears After the Text to Parameterize

In the **Appears after** section, you use the following options to specify which text, if any, should appear after the text output parameter:

Option	Description
Appears after (default)	Specifies that the information in the section appears after the text output parameter. To ignore the text that appears after the parameter, clear this check box.
Constant (default)	Displays the text that appears after the parameter.
Parameter	Sets the text string as a parameter.

Specifying What Appears Before the Text to Parameterize

In the **Appears before** section, you use the following options to specify which text, if any, should appear before the text output parameter:

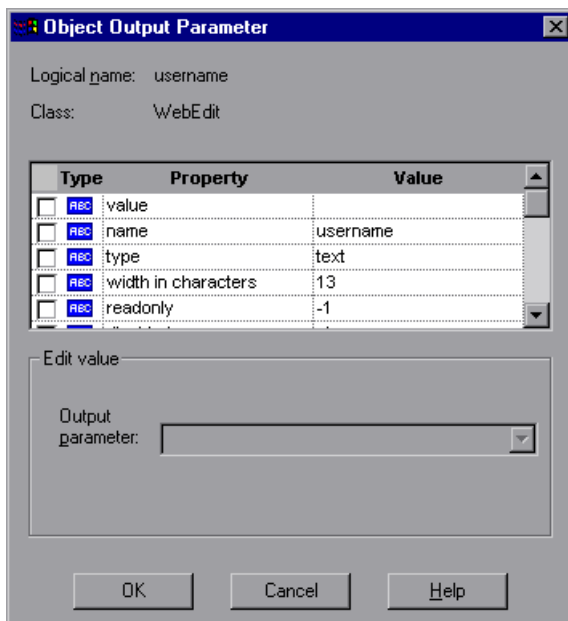
Option	Description
Appears before (default)	Specifies that the information in the section appears before the text output parameter. To ignore the text that appears before the parameter, clear this check box.

Option	Description
Constant (default)	Displays the text that appears before the parameter.
Parameter	Sets the text string as a parameter.

Parameterizing Object Output

You can parameterize an object on your Web page to create an object output parameter. To parameterize an object, you open the Object Output Parameter dialog box, as described in “Creating Output Parameters” on page 64.

In the Object Output Parameter dialog box, you can specify which property of the object to parameterize, and edit the values of this property.





Identifying the Object

The top part of the dialog box displays information about the object to parameterize:

Information	Description
Logical name	The name of the object as defined in the HTML code of the Web page.
Class	The type of object. In this example, the "WebEdit" class indicates that the object is an edit field.

Choosing which Property to Parameterize

The dialog box also displays the properties of the object you can parameterize, in a pane listing the properties, their values, and their types:

Option	Description
Check box	To parameterize a property, select the corresponding check box.
Type	The  icon indicates that the value of the property is a constant. The  icon indicates that the value of the property is a parameter.
Property	The name of the property to check.
Value	The value of the property to parameterize.

Choosing an Output Parameter

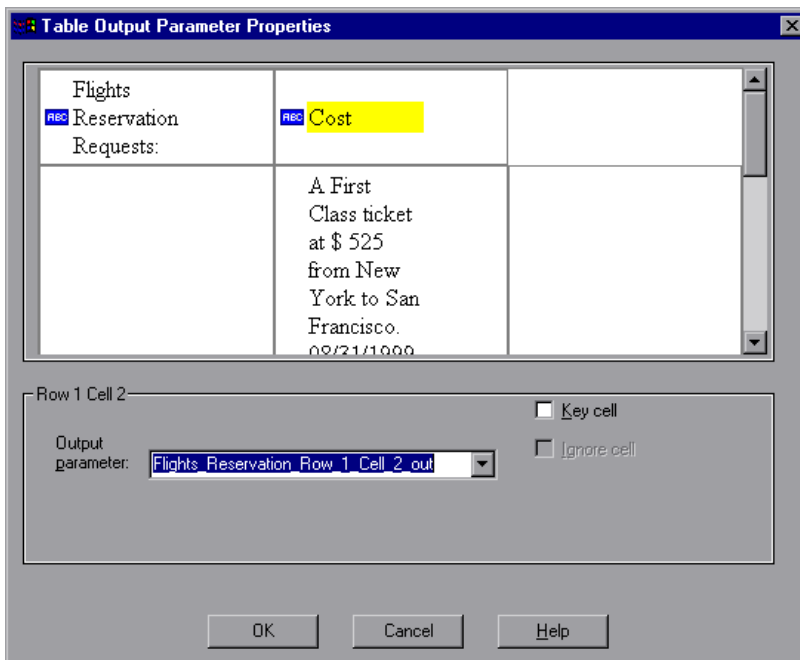
In the **Output Parameter** box, choose a name for the output parameter from the list, or enter a new name.

- To use a parameter that you already created, select it from the list.
- To create a new parameter, you can use the default parameter name, or type a descriptive name for the parameter.

Parameterizing Table Output

You can parameterize a text string in your table to create a table output parameter. To parameterize a text string in a table, you open the Table Output Parameter Properties dialog box, as described in “Creating Output Parameters” on page 64.

In the Table Output Parameter Properties dialog box, you can specify the name of the output parameter.



The dialog box displays rows and columns of a table. Your highlighted text string appears in a cell.

In the Output Parameter box, you can create a new output parameter or an existing output parameter.

- To use an output parameter that you already created, select an output parameter from the list.
- To create a new output parameter, you can use the default output parameter name, or type a descriptive name for the output parameter.

8

Using Regular Expressions

You can use regular expressions to increase the flexibility and adaptability of your tests. This chapter describes:

- ▶ Using Regular Expressions in Steps
- ▶ Using Regular Expressions in Object Checkpoints
- ▶ Using Regular Expressions in Text Checkpoints
- ▶ Regular Expression Syntax

About Regular Expressions

When you run your test, regular expressions enable to identify Web objects and text strings with varying values. You can use regular expressions when defining the properties of a step or when parameterizing a step, and when creating checkpoints with varying values. For example, when you create a checkpoint on a text string with a varying date, you can define the date as a regular expression.

A regular expression is a string that specifies a complex search phrase. By using special characters such as a period (.), asterisk (*), caret (^), and brackets ([]), you define the conditions of the search. When one of these special characters is preceded by a backslash (\), searches for the literal character.

Using Regular Expressions in Steps

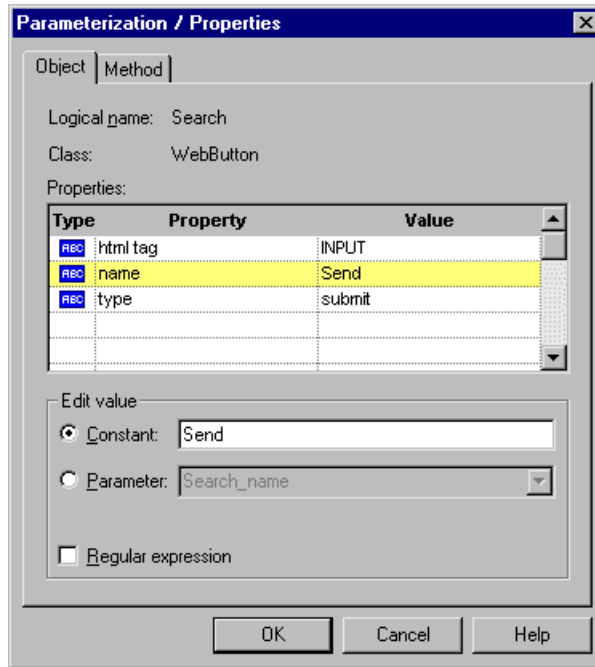
You can use regular expressions when defining or parameterizing a step in your test tree. A step is made up of an *object* that you navigate in your Web page, and/or a *method* by which you navigate the step. You can use regular expressions when defining or parameterizing the object of a step.

For example, your application may include a form in which the user inputs data and clicks the Send button to submit the form. When a required field is not completed, the form reappears for the user to complete. When resubmitting the form, the user clicks the Resend button. You can define the value of the object's "name" property as a regular expression, so that ignores variations in the object's property value.

To define a property value as a regular expression:

- 1 Right-click a step in your test tree. A menu opens.
- 2 Select **Parameterization/Properties**. The **Parameterization/Properties** dialog box opens.



3 Make sure the Object tab is selected.



The Object tab displays information about the object in the step:

Information	Description
Logical name	The name of the object as defined in the HTML code of the Web page.
Class	The type of object. In this example, the "WebButton" class indicates that the object is a button.

The Object tab displays the properties of the object in the step:

Option	Description
Type	The  icon indicates that the property value is a constant. The  icon indicates that the property value is a parameter.
Property	The name of the property value.
Value	The value of the property.

4 Click the property to set as a regular expression in the Properties section. The property is highlighted.

5 In the Edit value section, set the property value as a regular expression.

- To set the property value as a constant, click Constant.

In the Constant box, set the value as a regular expression. For information on regular expression syntax, see “Regular Expression Syntax,” on page 82.

- To set the property value as a parameter, click Parameter.

In the Parameter box, choose a parameter from the list or enter a new name: To use a parameter that you already created, select it from the list. To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter. For more information on parameterization, see Chapter 6, “Parameterizing Tests.”

Note: The property value in the Parameter box should not be defined as a regular expression. However, when you add additional values for the parameter into the data table, you can then specify the values as regular expressions.

For information on regular expression syntax, see “Regular Expression Syntax,” on page 82. For information on editing the data table, see Chapter 6, “Parameterizing Tests.”

- 6 Select the **Regular Expression** check box.
- 7 Click **OK** to save and close the dialog box.

If you created a new parameter, the **Astra parameters** dialog box prompts you to add the new parameter to the data table. Click **OK**. A new column is highlighted in the table for the new parameter.

In your test tree, the **P** icon next to the step indicates that the step has been parameterized.

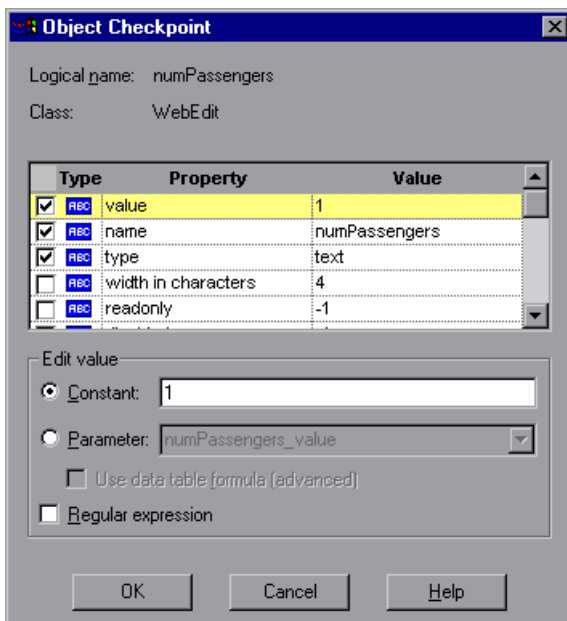
Using Regular Expressions in Object Checkpoints

When creating an object checkpoint to verify that an object appears on your Web application, you can also set the property value of the object as a regular expression, so that an object with varying names can be verified.

For example, suppose you want to check that when booking the number of passengers for a flight reservation in the “Mercury Tours” sample application, whole numbers are used. Astra LoadTest will ignore variations in the object’s property value as long as the value is a whole number.

To define a regular expression in an object checkpoint:

- 1 the Object Checkpoint dialog box.



The Object Checkpoint dialog box enables you to specify which properties of the object to check, and edit the values of these properties.

- 2 Select the check box of a property to be set as a regular expression. The property is highlighted.

3 In the **Edit value** section, set the property value as a regular expression.

- To set the property value as a constant, click **Constant**.

In the **Constant** box, set the value as a regular expression. For information on regular expression syntax, see “Regular Expression Syntax,” on page 82.

- To set the property value as a parameter, click **Parameter**.

In the **Parameter** box, choose a parameter from the list or enter a new name: To use a parameter that you already created, select it from the list. To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter. For more information on parameterization, see Chapter 6, “Parameterizing Tests.”



Note: The property value in the **Parameter** box should not be defined as a regular expression. However, when you add additional values for the parameter into the data table, you can then specify the values as regular expressions.

For information on regular expression syntax, see “Regular Expression Syntax,” on page 82. For information on editing the data table, see Chapter 6, “Parameterizing Tests.”

4 Select the **Regular Expression** check box.

5 Click **OK** to save and close the dialog box.

If you created a new parameter, the **Astra parameters** dialog box prompts you to add the new parameter to the data table. Click **OK**. A new column is highlighted in the table for the new parameter.

In your test tree, the  icon next to the step indicates that the step has been parameterized. The  icon indicates a checkpoint.

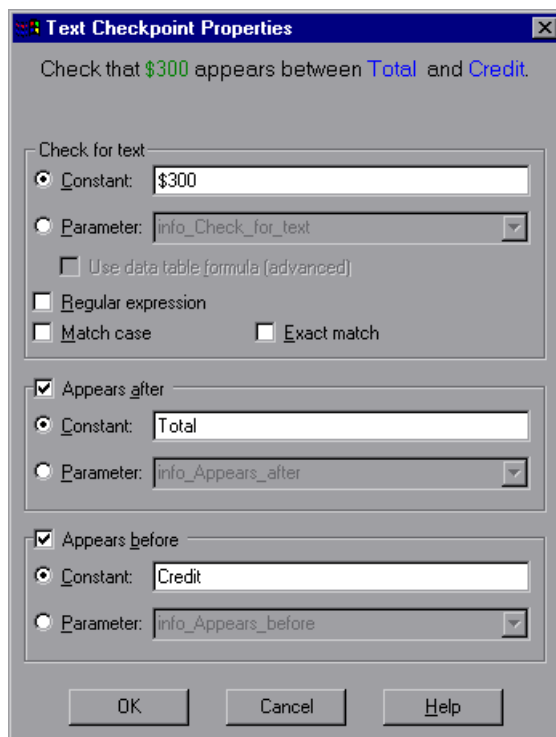
Using Regular Expressions in Text Checkpoints

When creating a text checkpoint to verify that a varying text string appears on your Web application, you define the text string as a regular expression.

For example, when booking a flight in the “Mercury Tours” sample application, the total cost charged to a credit card number should not be less than \$300. You define the amount as a regular expression, so that Astra LoadTest will ignore variations in the text string as long as the value is not less than \$300.

To define a regular expression in a text checkpoint:

- 1 the Text Checkpoint Properties dialog box.



The Text Checkpoint Properties dialog box enables you to specify which text to check as well as which text appears before and after the text to check.

2 In the **Check for text** section, define the text string as a regular expression.

- To set the text string as a constant, click **Constant**.

In the **Constant** box, define the text string as a regular expression. For information on regular expression syntax, see “Regular Expression Syntax,” on page 82.

- To set the text string as a parameter, click **Parameter**.

In the **Parameter** box, choose a parameter from the list or enter a new name: To use a parameter that you already created, select it from the list. To create a new parameter, either use the default parameter name or enter a descriptive name for the parameter. For more information on parameterization, see Chapter 6, “Parameterizing Tests.”



Note: The name in the **Parameter** box should not be defined as a regular expression. However, when you add additional values for the parameter into the data table, you can then specify the values as regular expressions.

For information on regular expression syntax, see “Regular Expression Syntax,” on page 82. For information on editing the data table, see Chapter 6, “Parameterizing Tests.”

3 Select the **Regular Expression** check box.

4 Click **OK** to save and close the dialog box.

If you created a new parameter, the **Astra parameters** dialog box prompts you to add the new parameter to the data table. Click **OK**. A new column is highlighted in the table for the new parameter.

In your test tree, the  icon next to the step indicates that the step has been parameterized. The  icon indicates a checkpoint.

Regular Expression Syntax

Astra LoadTest searches for all characters in a regular expression literally, except for a period (.), asterisk (*), caret (^), and brackets ([]), as described below. When one of these special characters is preceded by a backslash (\), Astra LoadTest searches for the literal character.

The following options can be used to create regular expressions:

Matching Any Single Character

A period (.) instructs Astra LoadTest to search for any single character. For example,

welcome.

matches welcomes, welcomed, or welcome followed by a space or any other single character. A series of periods indicates a sequence of unspecified characters.

Matching Any Single Character within a Range

In order to match a single character within a range, you can use square brackets ([]). For example, to search for a date that is either 1968 or 1969, write:

196[89]

You can use a hyphen (-) to indicate an actual range. For instance, to match any year in the 1960s, write:

196[0-9]

A hyphen does not signify a range if it appears as the first or last character within brackets, or after a caret (^).

A caret (^) instructs Astra LoadTest to match any character except for the ones specified in the string. For example:

```
[^A-Za-z]
```

matches any non-alphabetic character. The caret has this special meaning only when it appears first within the brackets.

Note that within brackets, the characters “.”, “*”, “[“ and “\” are literal. If the right bracket is the first character in the range, it is also literal. For example:

```
[]g-m]
```

matches the right bracket, and g through m.

Matching Specific Characters

An asterisk (*) instructs Astra LoadTest to match zero or more occurrences of the preceding character. For example:

```
Q*
```

causes Astra LoadTest to match Q, QQ, QQQ, etc.

9

Testing in Expert View

If you are an advanced user of Astra LoadTest, you can enhance your test in Expert View.

This chapter describes:

- ▶ Understanding Your Test Script Statements
- ▶ Programming in the Expert View
- ▶ Modifying Your Test Script

About Testing in Expert View

When you record a browser session, the Tree View tab graphically displays each step you perform in the form of a collapsible icon-based test tree. For every icon in the Tree View, the Virtual User Recorder displays a corresponding statement in the Expert View tab. The statements in Expert View are composed in Microsoft's VBScript language.

Understanding Your Test Script Statements

Unlike the icon-based step that appears in the Tree View, a step in the Expert View appears as a statement. For example, when a user clicks a button on a Web page, the following statement appears in the Expert View tab:

```
Browser ("Mercury Tours") .Page("Mercury Tours_4") .Frame("Info")  
.Image("reserveFlights") .Click 82,14
```

The following table explains the different components of the statement:

Item	Description
Browser ("Mercury Tours")	The logical name of the Web site is "Mercury Tours".
Page ("Mercury Tours_4")	The logical name of the page in the Web site is "Mercury Tours_4".
Image("reserveFlights")	The logical name of the button is "reserveFlights".
Click	The action performed on the button is "Click".
82, 14	The x- and y-coordinates of the mouse pointer when clicked on button.

Programming in the Expert View

The VBScript language consists of objects and methods. An *object* is a component of the Web application such as a button or a link. *Methods* are functions that are associated with particular objects. For example, the “click” method is associated with a button.

The list below contains a description of Astra LoadTest objects and their associated methods. These objects are specific to Astra LoadTest.

Object	Description	Methods
Browser	The browser	Close, Forward, FullScreen, Home, Navigate, Stop, Sync, Set
Page	The page	
Image	An image with or without a target URL	Click, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
Link	A text link	Click, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebArea	A client-side image map	Click, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebButton	A button	Click, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebCheckBox	A check box with "ON" and "OFF" states	Set, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebEdit	An edit field	Set, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute

Object	Description	Methods
WebFile	An edit field with a "Browse" button attached	Set, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebList	A dropdown or multi-selection list	Select, Deselect, ExtendSelect, Set, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebRadio	A radio button	Set, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
WebTable	A table	CellText, CellTextByContext, ChildItem, ChildItemCount, ColumnCount, RowCount, RowText, RowTextByContext, TextCellExist, TextRowExist, TextRowExist, Highlight, FireEvent, QueryValue, GetAttribute, SetAttribute
DataTable	A data table	Value, Get.CurrentRow, SetTableLine, GetRowsNumber, SetNextLine, SetPrevLine, SetFileName, GetFileName
Reporter	The object used to send information to the test report	ReportEvent

Sending Messages Using the Expert View

If you want to send a message during or following the test run, you can program which type of message to send in the Expert View.

The **Message** object contains the **Send** function for sending a message, stored in **Data**, with message level specified in **Level**. The syntax for **Message** is as follows:

```
Send(Data as String, [Level as Long])
```

The following table illustrates the message levels:

Level Number	Name	Explanation
1	Debug brief	Brief log
2	LrDebugResultData	Data returned by the server
6	LrMessageOutput	Messages sent by script to output window
7	LrMessageError	Error messages
8	LrDebugFullTrace	Full run-time trace
16	LrDebugExtended	Extended log
51	LrMessageStatus	Vuser status messages

Recording Your Data for Analysis

The **LRVB.LrDataPoint** object enables to record your own data for analysis. The recorded data is store in the **Data**, and the **Name** property specifies the name of the sample.

The syntax for using the **LRVB.LrDataPoint** object is as follows:

```
Data value As Double, [Name as String]
```

The following example illustrates the use of the **LRVB.LrDataPoint** object:

```
set lr = CreateObject ("LRVB.LrDataPoint")
lr. Data 40, "Joe"
```

Obtaining Environment Properties

The **LRVB.LrEnvironment** object enables you to obtain environment properties from the Controller.

The following list displays the environment properties you can access:

ControllerHostName	the name of the host machine running the Controller
GroupName	the current group
LocalHostName	the name of the host executing the Vuser script
ScenarioId	the current scenario
VuserId	the currently running Vuser

The following example illustrates the use of the **LRVB.LrEnvironment** object.

```
set lr = CreateObject (LRVB.LrEnvironment")
MsgBox lr.LocalHostName
```

Starting and Ending Transactions

The **Transaction** object enables you to start and end transactions.

The syntax for starting and ending transactions using the Transaction object is as follows:

```
StartT (Name As String)
EndT (Name As String, [Status As LrTransactionStatus])
    LrTransactionStatus
    2 - LrTransAuto
    1 - LrTransFail
    0 - LrTransPass
```

The following example illustrates the use of the Transaction object:

```
Transaction.StartT "trans1"
Transaction.EndT "trans1",2
```

Modifying Your Test Script

In Expert View mode, you can edit your test by typing directly into the test script. You can use the options in the menu bar and toolbar to modify your script. Note that if you make changes to the test script, the Virtual User Recorder makes corresponding changes to your test tree in the Tree View tab. If the Virtual User Recorder cannot understand the test script syntax when it switches back to the Tree View, it displays it as an error. In the Expert View, you can choose **Edit > Undo** to reverse the previous action.

10

Enhancing Your Tests with Programming

enhance your test using a few simple programming techniques.

This chapter describes:

- ▶ Inserting Functions
- ▶ Using Conditional Statements
- ▶ Sending Messages to Your Test Results
- ▶ Adding Comments

About Enhancing Your Tests with Programming

When recording, a test is generated by recording the typical processes that you perform on your Web application. As you navigate through your application, graphically displays each step you perform as an icon in a test tree.

Once you record your test, you can increase the power and flexibility of your tests by programming. The Virtual User Recorder includes the *Function wizard*, a programming tool which helps you to quickly and easily add recordable and non-recordable functions to your test. You can use the wizard to add functions that perform operations on Web objects or retrieve information from your application. For example, you can add a step that checks that an object exists, or you can retrieve the return value of a function.

The Virtual User Recorder also enables you to incorporate decision-making into your test. You can add conditional statements to control the logical flow of your test.

You can also define messages in your test that Astra LoadTest sends to your test results. To improve the readability of your tests, you can also add comments to your test.

This chapter introduces some programming concepts and shows you how to use simple programming techniques in order to create more powerful tests.

Inserting Functions

After recording, you can add additional functions to your tests using the Function wizard. With the wizard you can add recordable and non-recordable functions that perform operations on objects or retrieve information from your application. For example, the **QueryValue** function enables you to query the method argument value. You can use the return value of the function as an output parameter or as part of a conditional statement.

To insert a function in a test:

- 1 In the Tree View, right-click a step in the test tree. A menu opens. Choose **Insert > Step > Function**.

Tip: To add a function from the Expert View, click a statement in the test script. Right-click the highlighted object in the ActiveScreen. A menu opens. Choose **Insert Function**. The **Object Selection - Insert Function** opens. Select an object and click **OK**.

- 2 The **Function Wizard - Introduction** screen opens. Click **Next**.
- 3 The **Function Wizard - Step 1** screen opens.

Select a function and click **Next**.



- 4 If the function you chose returns a value, the **Function Wizard - Step 2** screen opens.

The following options are available:

Option	Description
Put the value in the data table (default)	Inserts the return value of the function as an output parameter into your data table. For more information, see Chapter 7, “Creating Output Parameters.”
Output parameter	Sets the name for the output parameter. You can accept the default name, select from the list, or enter a new name. For more information, see Chapter 7, “Creating Output Parameters.”
Place the value in a new condition statement	Inserts the return value of the function into a conditional statement. For more information, see “Using Conditional Statements” on page 97.

Click Next to continue.

If the function you chose has function arguments, the Function Wizard - Step 3 screen opens. The screen displays the arguments you can parameterize, in a pane listing the arguments, their values, and their types:

Option	Description
Type	The  icon indicates that the argument value is a constant. The  icon indicates that the argument value is a parameter.
Argument	The name of the argument whose value will be parameterized.
Value	The value of the argument to parameterize.

In the Edit value section, you use the following options to edit the argument value.

Option	Description
Constant (default)	Sets the argument value as a constant.
Parameter	Sets the argument value as a parameter. For more information, see Chapter 6, “Parameterizing Tests.”

Click Next to continue.

5 The Function Wizard - Finished screen opens.

Click Finish to complete the process and add the function to your test.

Using Conditional Statements

You can control the flow of your script with conditional statements. Using conditional statements, you can incorporate decision-making into your tests using *If...Then...Else* statements.

The *If...Then...Else* statement is used to evaluate whether a condition is true or false and, depending on the result, to specify one or more statements to run. Usually the condition is an expression that uses a comparison operator to compare one value or variable with another. The following comparison operators are available: *less than* $<$, *less than or equal to* $<=$, *greater than* $>$, *greater than or equal to* $>=$, *not equal* $<>$, and *equal* $=$.

Your *If...Then...Else* statement can be nested to as many levels as you need. It has the following syntax:

```
If condition Then statements [Else elsestatements ]
```

Or, you can use the block form syntax:

```
If condition Then
    [statements]
[Elseif condition-n Then
    [elseifstatements]] . . .
[Else
    [elsestatements]]
End If
```

Part of Statement	Description
condition	One or more expressions that evaluate to true or false. If the condition is null, it is treated as false.
statements	One or more statements, separated by colons, that are executed if the condition is true.
condition-n	Same as <i>condition</i> .
elseifstatements	One or more statements, separated by colons, that are executed if the associated <i>condition-n</i> is true.
elsestatements	One or more statements, separated by colons, that are executed if no previous <i>condition-n</i> expression is true.





For example, the statement below (as it appears in the Expert View) checks that the password edit box exists in the Mercury Tours application. **If** the edit box exists, **then** a password is entered; **else** a message is sent to test results.

```

If Browser("Mercury Tours").Page("Mercury Tours").Frame("navbar").WebEdit("password").Exist Then
    Browser("Mercury Tours").Page("Mercury Tours").Frame("navbar").WebEdit("password").Set "mercury"
Else
    Reporter.ReportEvent 2, "Password edit", "The edit field does not exists."
End If

```

In the test tree, the following icons are used to indicate the different levels of *If...Then...Else* statements:

Icon	Description
	Starts an <i>If</i> statement.
	Starts a <i>Then</i> statement.
	Starts an <i>Elseif</i> statement.
	Starts an <i>Else</i> statement.

To add a conditional statement:

- 1 In the Tree View, right-click a step in the test tree.
- 2 Choose **Insert > Step > If...Then...Else > If...Then**. The **Parameterization** dialog box opens.

Tip: You can also add a conditional statement by right-clicking a step in the test tree. A menu opens. Choose **Insert Step > If...Then...Else > If...Then**.

- 3 Set the expression as a constant or a parameter. Note that the expression must be a Boolean expression.
 - To set the expression as a constant, select **Constant**, and type the expression in the box. For example, type `i>5`.
 - To set the expression as a parameter, select **Parameter**, and choose a parameter from the list or enter a new parameter. For example, type `a>c` for a formula in a table. For more information, see Chapter 6, “Parameterizing Tests.”
- 4 Click **OK** to close the dialog box.
- 5 To complete the **Then** statement you can:
 - Record a new step and then use the **Cut/Paste** commands to add it to your **Then** statement.
 - Copy an existing step and paste it in your **Then** statement.
 - Click and drag a step to move it to your **Then** statement.
- 6 To nest an additional level to your statement, click the **Then** statement and choose one of the following options:

To add:	Choose:
an If statement	Insert > Step > If...Then...Else > If...Then
an Elseif statement	Insert > Step > If...Then...Else > Elseif...Then
an Else statement	Insert > Step > If...Then...Else > Else

To complete the new statement you can:

- Record a new step and then use the Cut/Paste commands to add it to your statement.
- Copy an existing step and paste it in your statement.
- Click and drag a step to move it to your statement.

Sending Messages to Your Test Results


You can define a message in your test that Astra LoadTest sends to your test results. For example, suppose you want to check that a password edit box exists in the Mercury Tours application. If the edit box exists, then a password is entered. Otherwise, Astra LoadTest sends a message to the test results indicating that the object is not found.

To send a message to your test results:

- 1 In the test tree, right-click a step. A menu opens.
- 2 Choose **Insert > Step > Report**. The **Insert Report** dialog box opens.
- 3 Select a status from the **Status** list.

Status	Description
Passed	sends a message if the step passes
Failed	sends a message if the step fails
General	sends a message regardless of the status of the step

- 4 In the **Name** box, type a step name to associate with the message. For example, “Password edit box”.
- 5 In the **Details** box, type a message to insert in your test results. For example, “Password edit box does not exist”.


After you run the test, the  icon in the Test Results window indicates a message.

Adding Comments

While programming, you can add comments to your tests. A *comment* is an explanatory remark in a program. When you run a test, Astra LoadTest does not process comments. Use comments to explain sections of a test in order to improve readability and to make tests easier to update.

To add a comment:

- 1 In the test tree, right-click a step. A menu opens.
- 2 Choose Insert > Step > Comment. The Insert Comment dialog box opens.
- 3 Type a comment and click OK.

A comment statement is added to your test. If you are working in Tree View, the  icon indicates a comment. In the Expert View, a comment is specified as *Rem*.

11

Running Tests in Stand-Alone Mode

In order to perform load testing with a recorded test, you use the Controller to run the test within a scenario. Before integrating the test into a scenario, you check its functionality by running the test in stand-alone mode.

Running the test in stand-alone mode means running the test without using the Controller. This is done to establish how the test will execute when run from the Controller.

This chapter describes:

- ▶ Running a Test
- ▶ Viewing the Results of a Test Run
- ▶ Printing Test Results

About Running a Test

When you run a test, the Virtual User Recorder navigates through your Web site, performing the steps you recorded. If the test does not contain parameterized values, the Virtual User Recorder runs the test once. If the test does contain parameters, the Virtual User Recorder runs the test for each row in the data table, using the parameters you specified.

Once the test run is complete, the Virtual User Recorder displays a report detailing the test results.

Running a Test

When you run a test, the Virtual User Recorder performs the steps you recorded on your Web site and displays each Web page in your browser.

To run a test:



- 1 If your test is not already open, choose **File > Open** or click the **Open** button to open the test.



- 2 Click the **Start Run** button on the toolbar, or choose **Run > Start**.

The Virtual User Recorder begins running the test. The Virtual User Recorder always runs a test from the first step in the test. As the Virtual User Recorder runs the test, it highlights each step in the test tree.

When the test stops running, the Test Results window opens.

Note: If you want to interrupt a test that is running, you can:



Click the **Pause** button or choose **Run > Pause**. The test pauses. To resume running a paused test, choose the **Start Run** button or choose **Run > Start**.



Click the **Stop** button or choose **Run > Stop**. The test stops running and the Test Results window opens.

Viewing the Results of a Test Run

After a test run, you can view the results in the Test Results window. By default, the Test Results window automatically opens when a test run is completed. For more information, see “Testing Options” on page 115.

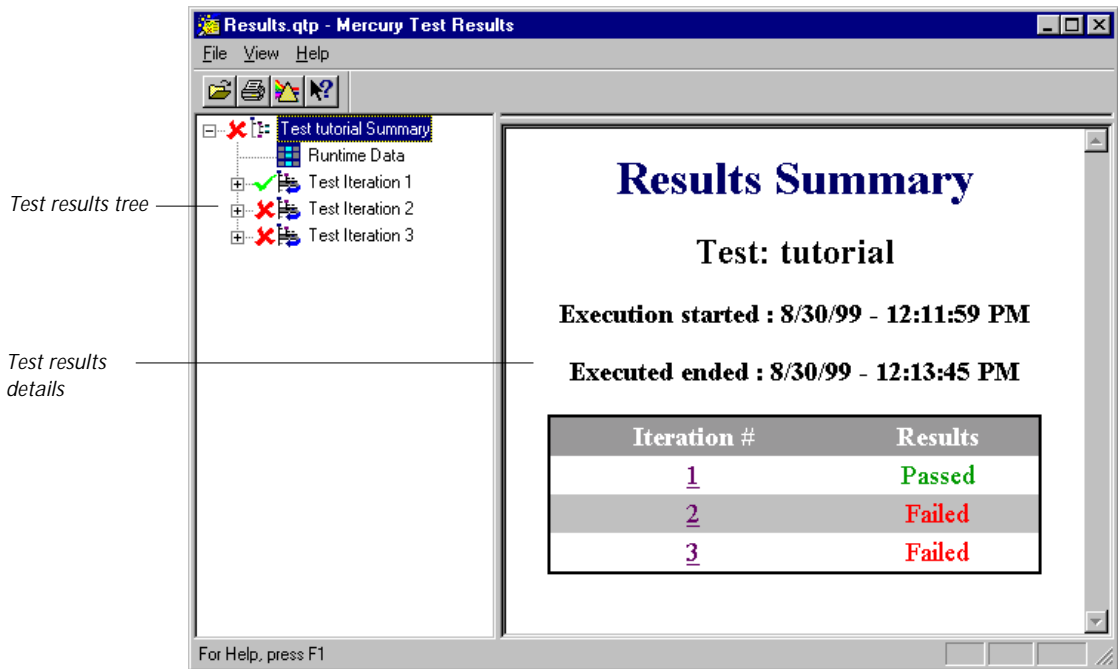
To view the results of a test run:

- 1 If the Test Results window is not already open then:



- ▶ Click the **Test Results** button or choose **Tools > Test Results**. The **Select Results File** dialog box opens.

- Select a results file (.qtp extension). Click Open. The Test Results window opens.



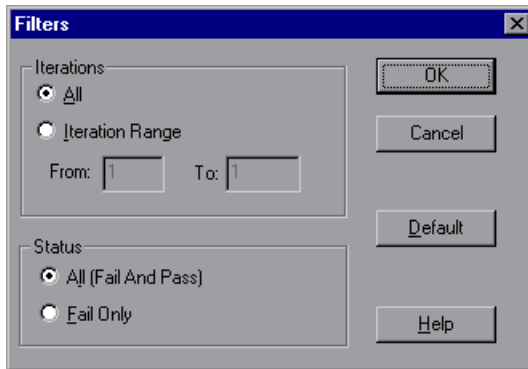
The Test Results window displays your test results. The left pane in the window displays the *test results tree*—a graphical representation of the test results. This includes a green check mark for a successful iteration, and a red X for a failed iteration. In the example, the tree includes three iterations. The right portion of the window displays the *test results details*—additional information for a selected branch of the report tree.

- 2 You can collapse or expand a branch in the test results tree in order to change the level of detail that the tree displays.
 - To collapse a branch, click the Collapse (-) sign to the left of the branch icon. The report tree hides the details for the branch and the Collapse sign changes to Expand.
 - To collapse all the branches in the report tree, choose View > Collapse All.

- To expand a branch, click the Expand (+) sign to the left of the branch icon. The tree displays the details for the branch and the Expand sign changes to Collapse.
- To expand all the branches in the report tree, choose **View > Expand All**.



- 3 To filter the information contained in your test results report, click the **Filter** button or choose **View > Filters**. The **Filters** dialog box opens.



The default filter options are displayed above.

- Click **Iteration Range** to limit the test results to a specified range of test iterations.
- Click **Fail Only** to limit the test results to test iterations that failed.



- 4 To view other test run results, click the **Open** button or choose **File > Open**. The **Open** dialog box opens. Select a results file (*.qtp* extension) and click the **Open** button.



- 5 To print the test results, see “**Printing Test Results**” on page 107.
- 6 Choose **File > Exit** to close the **Test Results** window.

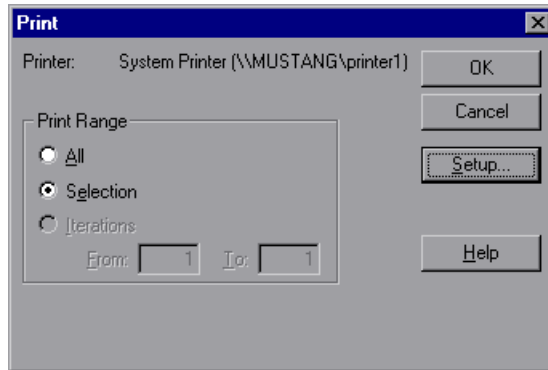
Printing Test Results

You can print your test results from the Test Results window.

To print the test results:



- 1 To print the report, click the **Print** button or choose **File > Print**. The **Print** dialog box opens.



- 2 Select a **Print Range** option:
 - Select **All** to print the entire results report.
 - Select **Selection** to only print a selected branch in the report tree.
 - Select **Iterations** to only print a specified range of test iterations.
- 3 Click **OK** to print.

12

Debugging Tests

Controlling test runs can help you to identify and eliminate defects in your tests.

This chapter describes:

- ▶ Using the Step Commands
- ▶ Pausing Test Runs
- ▶ Setting Breakpoints
- ▶ Deleting Breakpoints

About Debugging Test Scripts

After you create a test script you should check that it runs smoothly, without errors in syntax or logic. In order to detect and isolate defects in a test, you can use the Step and Pause commands to control how it runs. In addition, you can also control how the test runs by setting breakpoints.

The following Step commands are available:

- ▶ The Step Into command calls a function or displays another test.
- ▶ The Step Out command—used in conjunction with Step Into—completes the execution of a function or called test.
- ▶ The Step Over command executes a function or a called test.

You can also use the Pause command to temporarily suspend a test run. When you resume running the test, it continues from the point where you invoked the Pause command.

In addition, you can also control test runs by setting breakpoints. A breakpoint pauses a test run at a pre-determined point, enabling you to examine the effects of the test on your Web site.

Using the Step Commands

You can run a single line of a test using the Step Into, Step Out, and Step Over commands.



Step Into

Choose **Run > Step Into** or click the **Step Into** button to run only the current line of the active test script. If the current line of the active test calls another test or a function, the called test or function is executed in its entirety, and the called test or function is displayed in the Virtual User Recorder window.



Step Out

Choose **Run > Step Out** or click the **Step Out** button only after entering a test or a user-defined function using Step Into. Step Out runs to the end of the called test or user-defined function, returns to the calling test, and then pauses the test run.



Step Over

Choose **Run > Step Over** or click the **Step Over** button to run only the current step in the active test. When the current step calls another test or a user-defined function, the called test or function is executed in its entirety, but the called test script is not displayed in the Virtual User Recorder window.

Pausing Test Runs



You can temporarily suspend test runs by choosing **Run > Pause** or clicking the **Pause** button. A paused test stops running when all previously interpreted steps have been run.



To resume running a paused test, click the **Start Run** button or choose **Run > Start**. The test run continues from the point that you invoked the **Pause** command.

Setting Breakpoints

By setting a breakpoint you can stop a test run at a specific place in a test. A breakpoint is indicated by a red-colored hand in the left margin of the test window. pauses the test run when it reaches a breakpoint. You can examine the effects of the test run up to the breakpoint, make any necessary changes, and then continue running the test from the breakpoint.

You can use breakpoints to:

- suspend a test run and inspect the state of your application
- mark a point from which to begin stepping through a test script using the **Step** commands

To set a breakpoint:

- 1 Click a step or a line in the test where you want the test run to stop.



- 2 Choose **Debug > Toggle Breakpoint** or click the **Toggle Breakpoint** button. The breakpoint symbol appears in the left margin of the **Virtual User Recorder** window.

Note: The breakpoints you define are active only during your current session. If you terminate your session, you must redefine breakpoints to continue debugging the script in another session.

Deleting Breakpoints

You can delete a single breakpoint or all breakpoints defined for the current test using the Debug menu.



- To delete a single breakpoint, click a line in your test with the breakpoint symbol and choose **Debug > Toggle Breakpoint**, or click the **Toggle Breakpoint** button.

The breakpoint symbol is removed from the left margin of the window.



- To delete all breakpoints, choose **Debug > Clear All Breakpoints** or click the **Clear All Breakpoints** button.

All breakpoint symbols are removed from the left margin of the window.

13

Setting Testing Options

You can control how Astra LoadTest records and runs tests by setting testing options from the Test Settings dialog box.

This chapter describes:

- ▶ Setting Testing Options from the Test Settings Dialog Box
- ▶ Testing Options

About Setting Testing Options

The Virtual User Recorder testing options affect how you record and run tests. For example, you can set the speed at which Astra LoadTest runs a test, or set the timing-related settings used by Astra LoadTest.

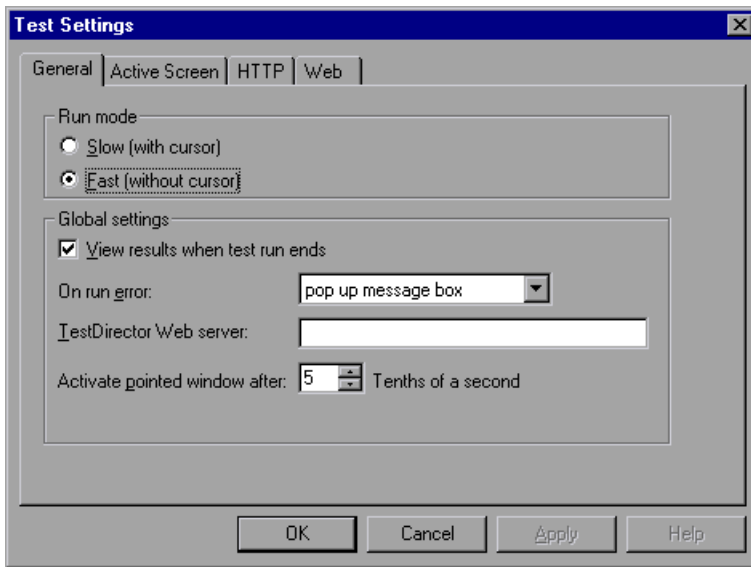
Setting Testing Options from the Test Settings Dialog Box

Before you record or run a test, you can use the Test Settings dialog box to modify your testing options. The values you set remain in effect for all tests.

To set testing options:

- 1 Choose Tools > Test Settings.

The Test Settings dialog box opens. It is divided by subject into four tabbed pages.



- 2 To choose a page, click a tab.
- 3 Set an option, as described in “Testing Options,” on page 115.
- 4 To apply your changes and keep the Test Settings dialog box open, click **Apply**.
- 5 When you are done, click **OK** to apply your changes and close the dialog box.

Testing Options

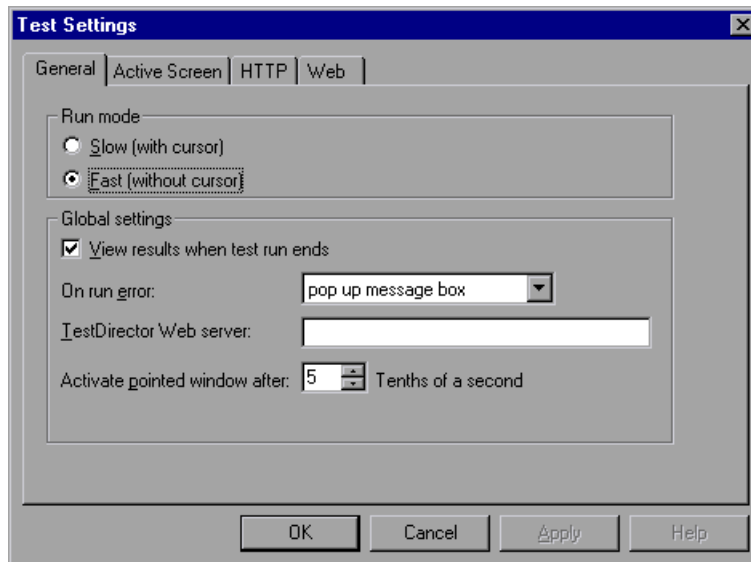
The Test Settings dialog box contains the following tabbed pages:

Tab Heading	Subject
General	Options for run mode and global settings
Active Screen	Options for displaying Web pages in the ActiveScreen tab in the Display pane
HTTP	Options to correlate HTTP values
Web	Options for recording tests

This section lists the testing options you can set using the Test Settings dialog box.

General Testing Options

The General tab options affect how Astra LoadTest runs tests, displays test results, and specifies the Web server for reporting defects.

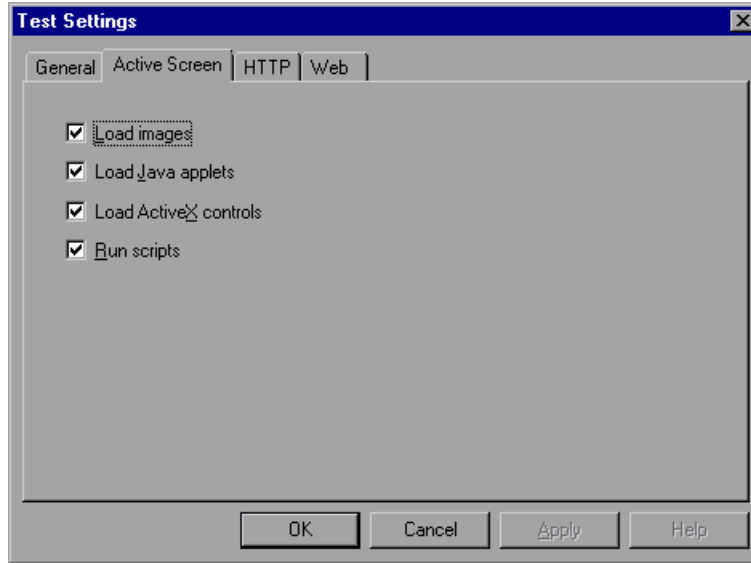


The General tab includes the following options:

Option	Description
Run mode - Slow (with cursor)	Instructs Astra LoadTest to run your test with the execution arrow in the left margin of the test, marking each step or statement as it is interpreted.
Run mode - Fast (without cursor)	Instructs Astra LoadTest to run your test without the execution arrow in the left margin of the test, marking each step or statement as it is interpreted.
View results when test run ends	Instructs Astra LoadTest to display the test results automatically following the test run.
On run error	Determines how Astra LoadTest responds to an error during a test run. Choose an option from the list: pop up message box displays an error message dialog box when an error occurs. proceed to next iteration jumps to the next iteration when an error occurs. stop run stops the test run when an error occurs.
TestDirector Web server	Designates the HTTP address to use to report and maintain defects in the Web Defect Manager. For example, if the TestDirector Web Defect Manager server is accessed using the <i>http://testdirector.mycompany.com/defects/bugs.htm</i> page, then the HTTP address should be specified as <i>http://testdirector.mycompany.com/defects/</i> .
Activate pointed window after	Specifies the time (in tenths of a second) that Astra LoadTest waits before it sets focus on the Web browser.

Active Screen Testing Options

The Active Screen tab options affect how the Virtual User Recorder displays Web pages in the ActiveScreen view in the Display pane.

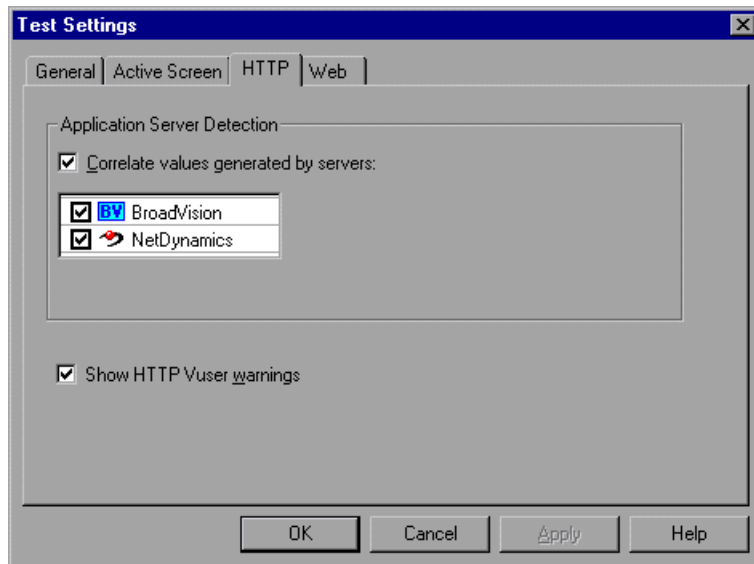


The Active Screen tab includes the following options:

Option	Description
Load images	Instructs Astra LoadTest to load images from your browser page to the ActiveScreen pane
Load Java applets	Instructs Astra LoadTest to load Java applets from your browser page to the ActiveScreen pane
Load ActiveX controls	Instructs Astra LoadTest to load ActiveX controls from your browser page to the ActiveScreen pane
Run scripts	Instructs Astra LoadTest to run scripts while loading your browser page on the ActiveScreen pane

HTTP Testing Options

The HTTP tab options affect how Astra LoadTest correlates dynamic information and issues HTTP warnings.

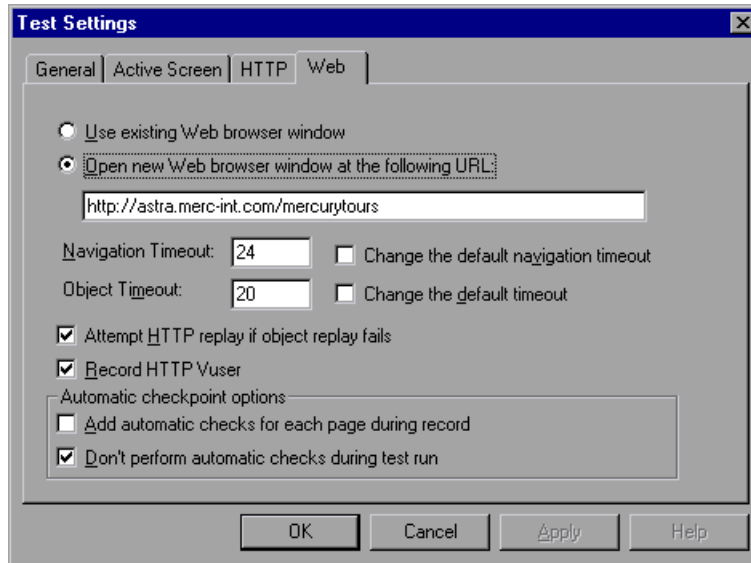


The Windows tab includes the following options:

Option	Description
Correlate values generated by servers:	Enables automatic detection of known Application Servers and correlates dynamic information to allow successful test runs.
Broadvision	Instructs Astra LoadTest to correlate values generated from a Broadvision server.
NetDynamics	Instructs Astra LoadTest to correlate values generated from a NetDynamics server.
Show HTTP Vuser warnings	Enables the suppression of HTTP Vuser warnings.

Web Testing Options

The Web tab options affect the recording options in the Virtual User Recorder.



The Web tab includes the following options:

Option	Description
Use existing Web browser window	Instructs Astra LoadTest to use your existing browser window to record a test.
Open new Web browser window at the following URL	Instructs Astra LoadTest to open a new browser session to record a test using the specified Web location address.
Navigation timeout	Indicates the interval (in seconds) Astra LoadTest waits for the Web page to load before running a test step.
Change the default navigation timeout	Changes the default Navigation timeout globally.

Option	Description
Object timeout	Indicates the interval (in seconds) Astra LoadTest waits to determine that the Web object is stable before running a test step.
Change the default timeout	Changes the default Object timeout globally.
Attempt HTTP replay if object replay fails	Instructs Astra LoadTest to substitute HTTP information during a Browser replay in the event of a failure during object replay.
Record HTTP Vuser	Instructs the Virtual User Recorder to record in HTTP mode. A test must be recorded in this mode for a HTTP replay to run.
Add automatic checks for each page during record	Instructs Astra LoadTest to add a checkpoint for each page navigated to in your site. By default the checkpoint checks links and image sources. You can also check for broken links.
Don't perform automatic checks during test run	Instructs Astra LoadTest to ignore the automatically added checkpoints when running your test.

Note: You can also set the **Use existing Web browser window** and **Open new Web browser window** at the following URL options for a specific test in the **Start Recording** dialog box, which opens when you start recording a new test.

Index

A

- ActiveScreen tab
 - in Display pane 9
 - in Test Settings dialog box 117
- application, sample 4
- ASCII 58
- Astra LoadTest
 - Data pane 6
 - Debug toolbar 12
 - Display pane 6
 - Load toolbar 12
 - menu bar 6
 - overview 5–14
 - Standard toolbar 11
 - starting 6
 - status bar 7
 - Test pane 6
 - Test toolbar 11
 - testing process 2
 - title bar 6
 - toolbar 6
 - window 6
- Astra LoadTest context sensitive help ix
- Astra LoadTest Readme file ix
- Astra LoadTest testing process 2
- Astra LoadTest Tutorial ix
- Astra LoadTest window 9
- AutoFill List command 61

B

- breakpoints
 - Clear All Breakpoints button 112
 - deleting 112
 - overview 109
 - setting 111
 - Toggle Breakpoint button 111, 112

- buttons, Astra LoadTest toolbar 10

C

- Check for Page dialog box 23
- checking
 - objects 32
 - pages 23
 - tables 34
 - text 29
- Checkpoint command 13
- checkpoints 21–34
 - Check for Page dialog box 23
 - checking objects 32
 - checking pages 23
 - checking tables 34
 - checking text 29
 - definition 21
 - modifying 34
 - Object Checkpoint dialog box 32
 - parameterizing 56–57
 - Text Checkpoint Properties dialog box 29
- Clear All Breakpoints button 112
- Clear All Breakpoints command 14
- Clear command 60
- Collapse All command 105
- commands using shortcut keys 13–14
- comments 102
- conditional statements 97
- Copy command 13, 60
- creating a new test 20
- creating checkpoints 21–34
- creating tests 15–20
- Currency(0) command 61
- Currency(2) command 61
- Cut command 13

D

- Data commands, Data Table tab 61
- Data pane, Astra LoadTest window 6, 10
 - Data Table tab 10
 - Runtime Data tab 10
- Data Table tab 10, 49
 - column definition 58
 - Data commands 61
 - Edit commands 60
 - editing 58–61
 - File commands 59
 - Format commands 61
 - importing data in ASCII 58
 - importing data in Microsoft Excel 95 58
 - importing data in Microsoft Excel 97 58
 - menu commands 59
 - row definition 58
- Debug toolbar, Astra LoadTest window 12
- debugging tests 109–112
 - deleting breakpoints 112
 - overview 109
 - pausing runs 111
 - setting breakpoints 111
 - Step Into button 110
 - Step Out button 110
 - Step Over button 110
- Delete command 13, 60
- deleting breakpoints 112
- Display pane, Astra LoadTest window 6, 9
 - ActiveScreen tab 9
 - Log tab 10

E

- Edit commands, Data Table tab 60
- Expand All command 106
- Expert View mode 4, 85–91
 - programming 87
 - viewing steps 86
- Expert View tab 9
- Export command 59

F

- File commands, Data Table tab 59
- Filter button 106
- Fixed command 61
- Format commands, Data Table tab 61
- Fraction command 61
- Function Wizard 93, 94–96

G

- General tab, Run-Time Settings dialog box 46
- General tab, Test Settings dialog box 115

H

- HTTP tab, Test Settings dialog box 118

I

- Insert command 60
- inserting
 - rendezvous points 37
- Iterations tab, Run-Time Settings dialog box 42

L

- load testing 35
 - rendezvous points 37
 - transactions 35
- Load toolbar 12
- Log tab 10
- Log tab, Run-Time Settings dialog box 40

M

- managing tests 20
 - creating new 20
 - opening 20
 - saving 20
- menu bar, Astra LoadTest window 6
- Mercury Interactive on the Web ix
- Mercury Tours sample application 4
- Microsoft Excel 95 58
- Microsoft Excel 97 58
- Microsoft Visual Basic Scripting language 4
- modifying checkpoints 34

N

- Network tab, Run-Time Settings dialog box 43
- New button 11
- New command 13

O

- Object Checkpoint dialog box 32
- Open Astra Test dialog box 17, 20
- Open button 11, 17, 20
- Open command 13
- opening tests 20
- Output messages 47
- Output Parameter command 13
- output parameters 63–72
 - creating 64–67
 - defined 63
 - objects 70–71
 - tables 72
 - text 68–70

P

- parameterization 49–61
 - checkpoints 56–57
 - data table 49
 - overview 49
 - steps 51–56
- parameterization (output) 63–72
 - objects 70–71
 - tables 72
 - text 68–70
- Parameterization/Properties command 13
- Parameterization/Properties dialog box 74
- Paste command 13, 60
- Paste Values command 60
- Pause button 12, 104, 111
- Pause command 14
- pausing test runs 111
- Percent command 61
- Performance tab, Run-Time Settings dialog box 44
- planning tests 16
- Print button 11, 107
- Print command 13

- Print Data Table command 59
- programming 93–102
 - adding functions to tests 94–96
 - comments 102
 - conditional statements 97
 - Function Wizard 93, 94–96
 - sending messages to test results 101

R

- recording tests 16–18
- regular expressions 73–83
 - defining in object checkpoints 78
 - defining in steps 74
 - defining in text checkpoints 80
 - overview 73
 - syntax 82
- Rename command 13
- rendezvous points 37
- resources
 - Astra LoadTest context sensitive help ix
 - Astra LoadTest Readme file ix
 - Astra LoadTest Tutorial ix
 - Mercury Interactive on the Web ix
 - support information ix
 - technical support online ix
- Run button 111
- running tests 103–107
 - Pause button 104
 - Start Run button 104
 - Stop button 104
 - Test Results button 104
 - Test Results window 105
 - viewing test results 104
- Runtime Data tab 10
- run-time settings 38
- Run-Time Settings dialog box
 - General tab 46
 - Iterations tab 42
 - Log tab 40
 - Network tab 43
 - Performance tab 44
 - User Information tab 45
- Run-time Settings dialog box 38

S

- sample application 4
- Save Astra Test dialog box 20
- Save button 11, 20
- Save command 13
- saving tests 20
- Scientific command 61
- sending messages
 - output 47
- setting
 - ActiveScreen options in the Test Settings dialog box 117
 - breakpoints 111
 - General options in the Run-Time Settings dialog box 46
 - General options in the Test Settings dialog box 39, 115
 - HTTP options in the Test Settings dialog box 118
 - Iteration options in the Run-Time Settings dialog box 42
 - Log options in the Run-Time Settings dialog box 40
 - Network options in the Run-Time Settings dialog box 43
 - Performance options in the Run-Time Settings dialog box 44
 - run-time options 38
 - testing options 113–120
 - User Information options in the Run-Time Settings dialog box 45
 - Web options in the Test Settings dialog box 119
- shortcut keys 13–14
- Sort command 61
- Standard toolbar, Astra LoadTest window 11
- Start command 13
- Start Record button 17
- Start Recording dialog box 17
- Start Run button 104
- status bar, Astra LoadTest window 7
- Step commands 110
- Step Into button 12, 110
- Step Into command 13, 14
- Step Out button 12, 110
- Step Out command 14

- Step Over button 12, 110
- Stop button 104
- Stop command 14
- Stop Record button 18
- support information ix

T

- technical support online ix
- test
 - checkpoints 21–34
 - creating new 20
 - debugging 109–112
 - managing 20
 - opening 20
 - parameterization 49–61
 - parameterization (output) 63–72
 - pausing runs 104, 111
 - planning 16
 - printing results 107
 - programming 93–102
 - recording 16–18
 - running 103–107
 - saving 20
 - viewing test results 104
- Test pane, Astra LoadTest window 6, 8
 - Expert View tab 9
 - Tree View tab 8
- Test Results button 104
- Test Results window 105
 - test results details 105
 - test results tree 105
- Test Settings dialog box 114
 - ActiveScreen tab 117
 - General tab 115
 - HTTP tab 118
 - Web tab 119
- Test toolbar, Astra LoadTest window 11
- test tree
 - creating 17
 - definition 18
- testing
 - load 35–48
- testing in Expert View 85–91
- testing process 2
 - analyzing test results 4

- creating tests 2
- running tests 3
- Text Checkpoint Properties dialog box 29, 80
- title bar, Astra LoadTest window 6
- Toggle Breakpoint button 111, 112
- Toggle Breakpoint command 14
- toolbar, Astra LoadTest window 6
- toolbars, Astra LoadTest window
 - Debug 12
 - Load 12
 - Standard 11
 - Test 11
- transactions 35
 - inserting 36
- Tree View tab 8

U

- User Information tab, Run-Time Settings dialog box 45

V

- viewing test results 104

W

- Web tab, Test Settings dialog box 119



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.merc-int.com

Customer Support: web.merc-int.com



ALTUG3. 0/ 02