HP LoadRunner

for the Windows operating systems

Software Version: 9.50

Tutorial

Manufacturing Part Number: T7182-90018

Document Release Date: January 2009

Software Release Date: January 2009

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Third-Party Web Sites

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

Copyright Notices

Trademark Notices

Documentation Updates

This guide's title page contains the following identifying information:

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

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

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

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

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

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

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

Support

You can visit the HP Software Support web site at:

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

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Table of Contents

Welcome to the LoadRunner Tutorial

Welcome to the LoadRunner Tutorial, a self-paced printable guide, designed to lead you through the process of load testing and familiarize you with the LoadRunner testing environment.

The first lesson provides an introduction to LoadRunner and testing concepts.

Lesson 2 provides a sample test that illustrates the power of the LoadRunner tool.

Lessons 3 through 6 describe how to create a script using the HP Virtual User Generator.

Lessons 7 through 9 explain how to design and run load tests using the LoadRunner Controller.

Lesson 10 introduces the Analysis tool, showing you how to create graphs and reports which will help you analyze your load test.

At the conclusion of this tutorial, you will be ready to design, run, and monitor a simple test on your own application. It is recommended that you move through the tutorial in the order in which the information is presented.

1

Introducing LoadRunner

Over the last 20 years, companies have turned to software as a means of automating work. Software applications have been used to drive huge efficiency and productivity gains and to provide a new medium for collaboration and information sharing in a global economy. Software applications have, in fact, become the primary channel both for business-critical information sharing and transaction processing of all kinds. Today, software applications—from e-mail to CRM to Transaction Processing—**are** the business.

While software development technologies have changed and matured tremendously in this time period, the complexity of modern applications has exploded. Applications may utilize tens and hundreds of components to do work once done with paper or by-hand. There is a direct correlation between the degree of application complexity and the number of potential points of failure in a business process. This makes it increasingly difficult to isolate the root cause of a problem.

Moreover, software applications are not like cars. They do not have permanent parts that are replaced only when they wear out. Whether to deliver competitive advantage or to respond to changes in business conditions, software applications change weekly, monthly, and yearly. This stream of change introduces yet another set of risks that companies have to manage.

The incredible pace of change and the explosion of software complexity introduce tremendous risk into the software development process. Rigorous performance testing is the most common strategy to both quantify and reduce this risk to a business. Automated load testing with HP LoadRunner is an essential part of the application deployment process.

Why should you automate performance testing?

Automated Performance Testing is a discipline that leverages products, people, and processes to reduce the risks of application, upgrade, or patch deployment. At its core, automated performance testing is about applying production workloads to pre-deployment systems while simultaneously measuring system performance and end-user experience. A well-constructed performance test answers questions such as:

- Does the application respond quickly enough for the intended users?
- Will the application handle the expected user load and beyond?
- Will the application handle the number of transactions required by the business?
- Is the application stable under expected and unexpected user loads?
- Are you sure that users will have a positive experience on go-live day?

By answering these questions, automated performance testing quantifies the impact of a change in business terms. This in turn makes clear the risks of deployment. An effective automated performance testing process helps you to make more informed release decisions, and prevents system downtime and availability problems.

What are the LoadRunner components?

LoadRunner contains the following components:

- The **Virtual User Generator** captures end-user business processes and creates an automated performance testing script, also known as a Vuser script.
- The **Controller** organizes, drives, manages, and monitors the load test.
- The **Load Generators** create the load by running Vusers.
- **Analysis** helps you view, dissect, and compare the performance results.
- The **Launcher** provides a single point of access for all of the LoadRunner components.

Understanding LoadRunner Terminology

Scenario A **scenario** is a file that defines the events that occur during each testing session, based on performance requirements.

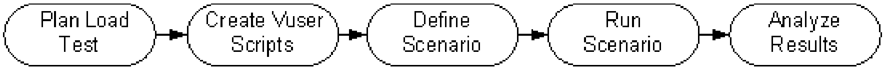
Vusers In the scenario, LoadRunner replaces human users with **virtual users** or **Vusers.** Vusers emulate the actions of human users working with your application. A scenario can contain tens, hundreds, or even thousands of Vusers.

Vuser Script The actions that a Vuser performs during the scenario are described in a **Vuser Script**.

Transactions To measure the performance of the server, you define **transactions**. A transaction represents end-user business processes that you are interested in measuring.

What is the load testing process?

Load testing typically consists of five phases: planning, script creation, scenario definition, scenario execution, and results analysis.



- **Plan Load Test.** Define your performance testing requirements, for example, number of concurrent users, typical business processes, and required response times.
- **Create Vuser Scripts.** Capture the end-user activities into automated scripts.
- **Define a Scenario.** Use the **LoadRunner Controller** to set up the load test environment.
- **Run a Scenario.** Drive, manage, and monitor the load test from the **LoadRunner Controller**.
- **Analyze the Results.** Use **LoadRunner Analysis** to create graphs and reports, and evaluate the performance.

Getting Familiar with HP Web Tours

To illustrate the HP solution, this tutorial uses sample performance requirements for a sample application. This application, HP Web Tours, is a Web-based travel agency system. HP Web Tours users connect to a Web server, search for flights, book flights, and check flight itineraries.

While LoadRunner supports over 40 types of applications, this tutorial demonstrates load testing Web-based applications. If you are load testing applications that are not Web-based, please contact HP for assistance.

In this section, you will become familiar with the HP Web Tours application.

Opening HP Web Tours

You will use the HP Web Tours application to experience hands-on performance testing. Before proceeding with the tutorial, follow these steps to familiarize yourself with the look and feel of the application.

1 **Make sure that the sample Web server is running.**

The Web server automatically starts after the LoadRunner installation and reboot. If you rebooted your system again and the server is not running, select **Start** > **All Programs** > **LoadRunner** > **Samples** > **Web** > **Start Web Server**.

Note: If you try to start the Web server and it is already running, an error message appears. You can disregard the message and continue with the tutorial.

2 Open the HP Web Tours application.

Select **Start** > **All Programs** > **LoadRunner** > **Samples** > **Web** > **HP Web Tours Application**. A browser opens with the HP Web Tours home page.

Note: Ensure that LoadRunner is installed in the default directory on your computer. If LoadRunner is installed in a non-default directory, the HP Web Tours application will not open.

3 Log on to HP Web Tours.

Type in the following information:

- **User name:** jojo
- **Password:** bean

On the left pane, click **login**. HP Web Tours welcomes you to the application.

4 Reserve a flight.

On the left pane click **Flights**. The Find Flight page opens. Change the **Arrival city** to Los Angeles. Click **Continue**.

5 End your HP Web Tours session.

On the left pane, click **Sign Off** to log off.

Application Requirements

Now that you are familiar with HP Web Tours, imagine that you are the performance engineer responsible for signing off that the application meets the needs of your business. Your project manager has given you 4 criteria for release:

1. HP Web Tours must successfully handle 10 concurrent travel agents.
2. HP Web Tours must be able to process 10 simultaneous flight bookings with response time not exceeding 90 seconds.
3. HP Web Tours must be able to handle 10 travel agents running simultaneous itinerary checks with response time not exceeding 120 seconds.
4. HP Web Tours must be able to handle 10 agents signing in and signing out of the system with response time not exceeding 10 seconds.

The tutorial will walk you through the process of building load tests that validate each business requirement so that you can attach a pass or fail before release.

2

The Power of LoadRunner

To illustrate the power of LoadRunner, you will run and analyze a load test against a database application, with up to 10 concurrent users. The test will emulate travel agents simultaneously using the flight reservation system (for example, logging on, searching flights, purchasing flights, checking itineraries, and logging off).

During the test, you will observe how the Web server behaves under load using LoadRunner's online monitors. In particular, you will see how an increase in load affects the time that it takes the server to respond to a user action (transaction response time) and causes errors.

After you have seen how LoadRunner can be used to generate load on a system and measure the responsiveness of the system to that load, you will learn how to use the LoadRunner components (VuGen, Controller, and Analysis) to create and run your own test, and analyze test results.

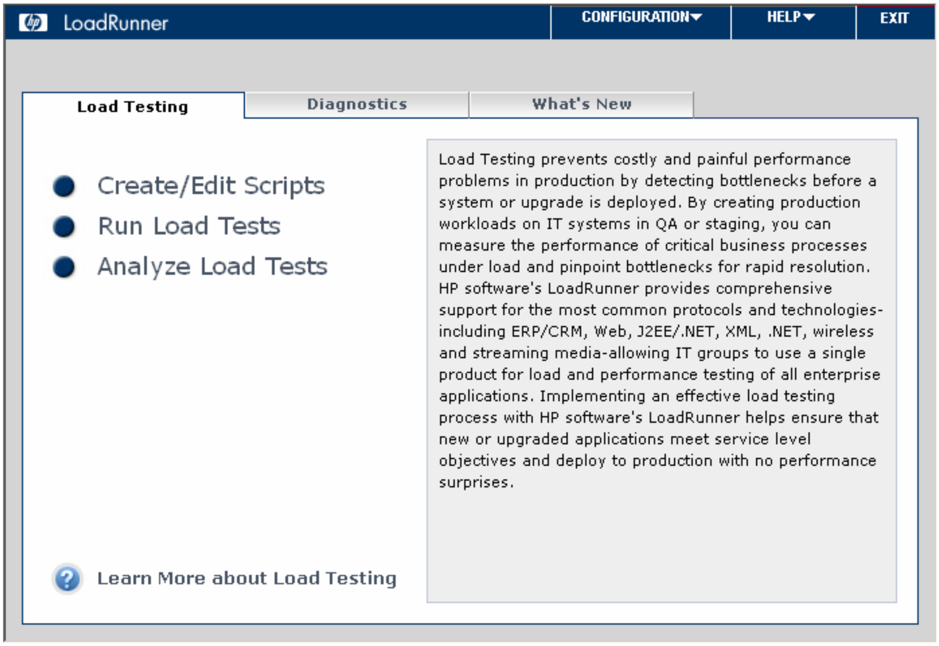
Note: LoadRunner allows you to purchase a license to use as many Vusers as you need to effectively test your application. For the purposes of this LoadRunner trial version, however, you are licensed to use 10 Vusers.

Creating the Load Test

The Controller is the central console from which you build, manage, and monitor your test. You will use the Controller to run a sample script that emulates the actions performed by a real user, and create load on the system by having a number of Vusers concurrently perform these actions.

1 Open the HP LoadRunner window.

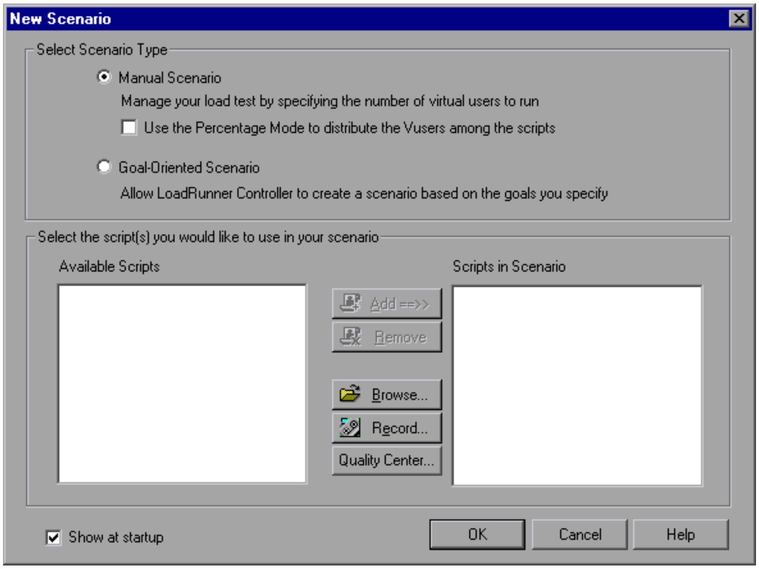
Select **Start** > **All Programs** > **LoadRunner** > **LoadRunner**. The HP LoadRunner Launcher window opens.



Load Testing prevents costly and painful performance problems in production by detecting bottlenecks before a system or upgrade is deployed. By creating production workloads on IT systems in QA or staging, you can measure the performance of critical business processes under load and pinpoint bottlenecks for rapid resolution. HP software's LoadRunner provides comprehensive support for the most common protocols and technologies-including ERP/CRM, Web, J2EE/.NET, XML, .NET, wireless and streaming media-allowing IT groups to use a single product for load and performance testing of all enterprise applications. Implementing an effective load testing process with HP software's LoadRunner helps ensure that new or upgraded applications meet service level objectives and deploy to production with no performance surprises.

2 Open the Controller.

In the **Load Testing** tab, click **Run Load Tests**. By default, the LoadRunner Controller opens with the New Scenario dialog box.



New Scenario

Select Scenario Type

● Manual Scenario

Manage your load test by specifying the number of virtual users to run

□ Use the Percentage Mode to distribute the Vusers among the scripts

○ Goal-Oriented Scenario

Allow LoadRunner Controller to create a scenario based on the goals you specify

Select the script(s) you would like to use in your scenario

Available Scripts

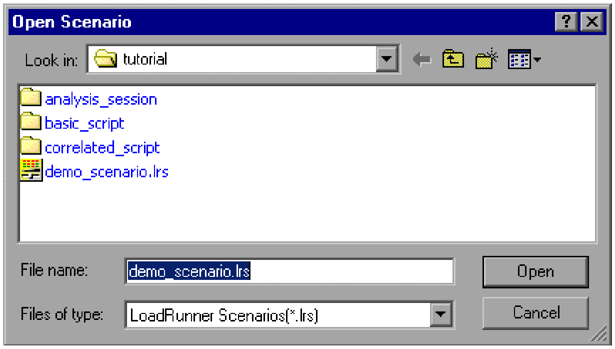
Scripts in Scenario

☑ Show at startup

Click **Cancel**.

3 Open the sample test.

In the Controller menu, select **File** > **Open**, and open **demo_scenario.lrs** in the <LoadRunner Installation>**Tutorial** directory.



Open Scenario

Look in: tutorial

analysis_session
basic_script
correlated_script
demo_scenario.lrs

File name: demo_scenario.lrs

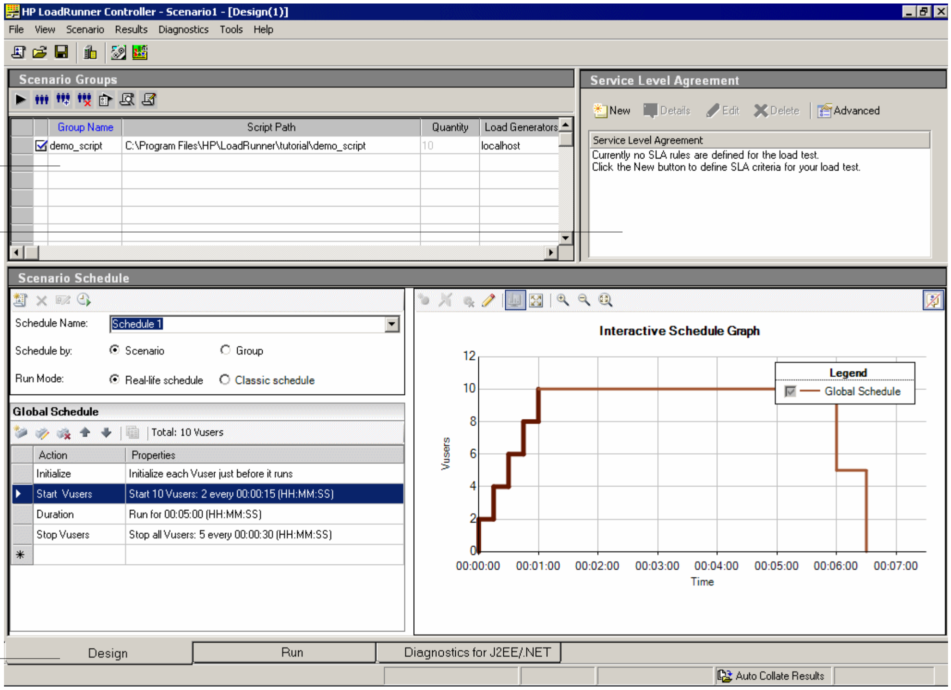
Files of type: LoadRunner Scenarios(*.lrs)

The Design tab of the LoadRunner Controller opens, and the **demo_script** test appears in the Scenario Groups pane. You can see that 10 Vusers have been assigned to run the test.

Scenario Groups section

SLA

Design Tab



Group Name	Script Path	Quantity	Load Generators
☑ demo_script	C:\Program Files\HP\LoadRunner\tutorial\demo_script	10	localhost

Service Level Agreement
Currently no SLA rules are defined for the load test.
Click the New button to define SLA criteria for your load test.

Schedule Name: Schedule 1

Schedule by: ● Scenario ○ Group

Run Mode: ● Real-life schedule ○ Classic schedule

Global Schedule

Total: 10 Vusers

Action	Properties
Initialize	Initialize each Vuser just before it runs
Start Vusers	Start 10 Vusers: 2 every 00:00:15 (HH:MM:SS)
Duration	Run for 00:05:00 (HH:MM:SS)
Stop Vusers	Stop all Vusers: 5 every 00:00:30 (HH:MM:SS)

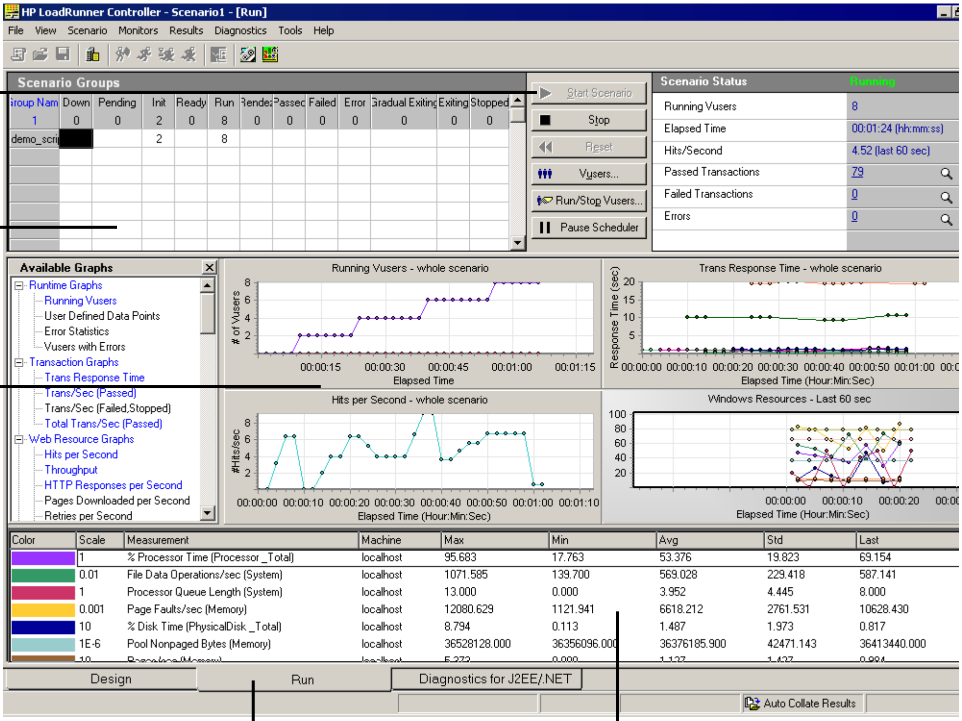
Note: If you did not install the tutorial to the default LoadRunner installation directory, the script path will be incorrect (indicated by the script path being displayed in red). To enter the correct path, select the script and click the down arrow. Click the **Browse** button and navigate to **demo_script** in the <LoadRunner Installation>**Tutorial** directory, and then click **OK**.

You are now ready to run the test.

Running the Load Test

In the **Run** tab, click the **Start Scenario** button. The Controller run view is displayed, and the Controller begins running the scenario.

In the Scenario Groups pane, you can watch Vusers gradually start to run and generate load on the system. You can see the responsiveness of the server to the Vuser actions in the online graphs.



Group Nam	Down	Pending	Init	Ready	Run	Rende:	Passec	Failed	Error	Gradual Exiting	Exiting	Stopped
1	0	0	2	0	8	0	0	0	0	0	0	0
demo_scri			2		8							

Scenario Status	Running
Running Vusers	8
Elapsed Time	00:01:24 (hh:mm:ss)
Hits/Second	4.52 (last 60 sec)
Passed Transactions	79
Failed Transactions	0
Errors	0

Color	Scale	Measurement	Machine	Max	Min	Avg	Std	Last
🟪	1	% Processor Time (Processor _Total)	localhost	95.683	17.763	53.376	19.823	69.154
🟩	0.01	File Data Operations/sec (System)	localhost	1071.585	139.700	569.028	229.418	587.141
🟥	1	Processor Queue Length (System)	localhost	13.000	0.000	3.952	4.445	8.000
🟨	0.001	Page Faults/sec (Memory)	localhost	12080.629	1121.941	6618.212	2761.531	10628.430
🟦	10	% Disk Time (PhysicalDisk _Total)	localhost	8.794	0.113	1.487	1.973	0.817
🟦	1E-6	Pool Nonpaged Bytes (Memory)	localhost	36528128.000	36356096.000	36376185.900	42471.143	36413440.000
🟫	10	Pages/sec (Memory)	localhost	5.272	0.000	1.127	1.427	0.994

Start Scenario

Scenario Groups section

Online Graphs area

Run Tab

Graph Measurements area

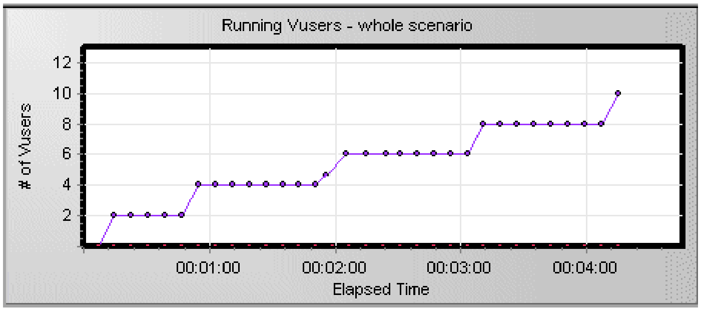
Monitoring the Load Test

While creating load on an application, you want to see how the application performs in real time and where potential bottlenecks exist. You use LoadRunner's suite of integrated monitors to measure the performance of every single tier, server, and component of the system during the load test. LoadRunner includes monitors for a variety of major backend system components including Web, application, network, database, and ERP/CRM servers.

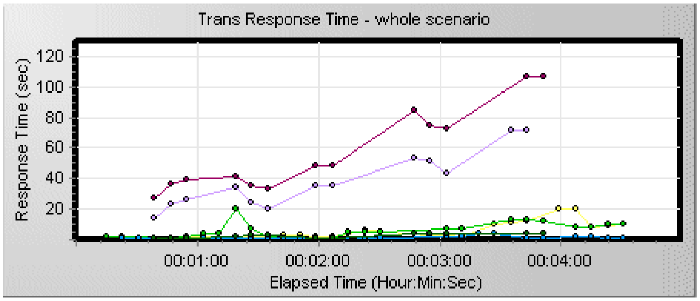
1 View the default graphs.

By default, the Controller displays the Running Vusers, Transaction Response Time, Hits per Second, and Windows Resources graphs. The first three do not require configuration. The Windows Resources monitor has been configured for you for this test.

➤ **Running Vusers - whole scenario** graph lets you monitor the number of Vusers that are running at a given time. You can see that the Vusers gradually start running at a rate of 2 Vusers every 1 minute.



- **Transaction Response Time - whole scenario.** This graph lets you monitor the amount of time it takes for each transaction to be completed. You can see how long it takes for your customers to log on, search for flights, purchase flights, check itineraries, and log off the system.



You can see that as more and more Vusers work on the application under test, transaction response times increase and the level of service provided to the customer degrades.

- **Hits per Second - whole scenario.** This graph lets you monitor the number of hits (HTTP requests) on the Web server made by Vusers during each second of the scenario run. This enables you to follow the amount of load that is generated on the server.

- **Windows Resources.** This graph lets you monitor the Windows resource usage measured during a scenario (such as CPU, disk, or memory utilization). You will learn how to configure Windows Resources and other monitors in Lesson 7, “Creating a Load Testing Scenario.”

Tips:

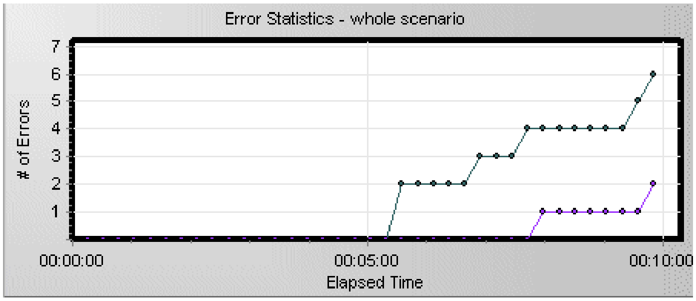
- Each measurement appears on a color-coded row in the graph legend. Each row corresponds to a line in the graph with the same color.

- Selecting a row highlights the corresponding line in the graph, and vice versa.

2 View error information.

If your machine is handling a heavy load, you may encounter errors.

In the Available Graphs tree, select the **Errors Statistics** graph, and drag it into the Windows Resources graph pane. The Error Statistics graph provides details about the number of errors and the time at which they occurred during the scenario. Errors are grouped by source—for example, the location in the script or the load generator name.



In this example, you can see that after 5 minutes, the system started to encounter an increasing number of errors. These errors were caused by timeouts due to degrading response times.

Note: It will take your scenario a few minutes to run. You can go back and forth from the graphs to the Vusers as your scenario proceeds to display your results online.

Analyzing Results

At the conclusion of the test run, LoadRunner provides in-depth analysis composed of detailed graphs and reports. You can compare multiple graphs by combining results from several scenarios. Alternatively, you can use the Auto-correlate tool to merge all the graphs that contain data that could have had an effect on the response time, to pinpoint what was happening at the moment the problem occurred. Using these graphs and reports, you can easily identify the bottlenecks in your application, and determine what changes need to be made to your system to improve its performance. You will learn how to use the Analysis tool in Lesson 10, “Analyzing Your Scenario.”

You can open Analysis to view the scenario results by selecting **Results** > **Results Settings** or clicking the **Analyze Results** button. The results are saved to the *<LoadRunner installation>***Results****tutorial_demo_res** directory.

Where To Go From Here

Now that you have discovered the power of LoadRunner’s automated testing process, you are ready to learn how to build a load-testing script, design and run your own test, and analyze the test results. The following lessons will walk you through the process of achieving these results.

3

Building Scripts

To create load, you first build automated scripts that emulate real user behavior.

In this lesson you will cover the following topics:

- Introducing the Virtual User Generator (VuGen)
- How do I start recording user activities?
- Using VuGen Wizard Mode
- How do I record a business process to create a script?
- How do I view the script?

Introducing the Virtual User Generator (VuGen)

In a testing environment, LoadRunner replaces human users at physical machines with **Vusers**. Vusers create load on a system by emulating actions of typical users in a repeatable and predictable manner.

The LoadRunner Virtual User Generator (VuGen) works on a record-and-playback principle. As you walk through a business process on your application, VuGen records your actions into automated scripts which will form the foundation of your load tests.

Note: If you have completed the HP LoadRunner Quick Start, you will notice that the script steps that you recorded there are identical to those that you will record in the following section. However, the overall recording process is explained here in greater detail.

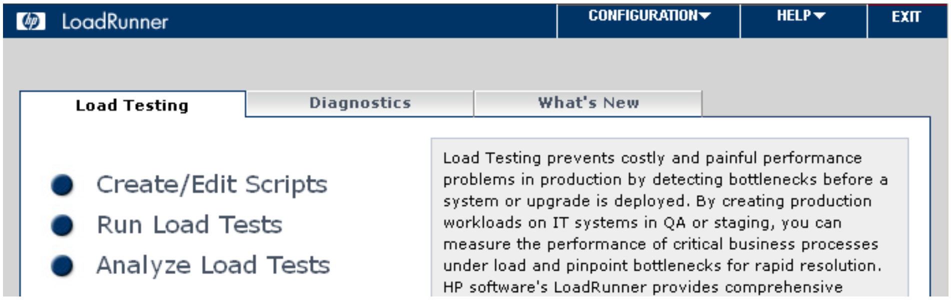
How do I start recording user activities?

To begin recording user actions, you open VuGen and create a blank script. You fill the blank script by recording events and adding manual enhancements.

In this section, you will open VuGen and create a blank Web script.

1 Start LoadRunner.

Select **Start** > **All Programs** > **LoadRunner** > **LoadRunner**. The HP LoadRunner Launcher window opens.



Load Testing prevents costly and painful performance problems in production by detecting bottlenecks before a system or upgrade is deployed. By creating production workloads on IT systems in QA or staging, you can measure the performance of critical business processes under load and pinpoint bottlenecks for rapid resolution. HP software's LoadRunner provides comprehensive

2 Open VuGen.

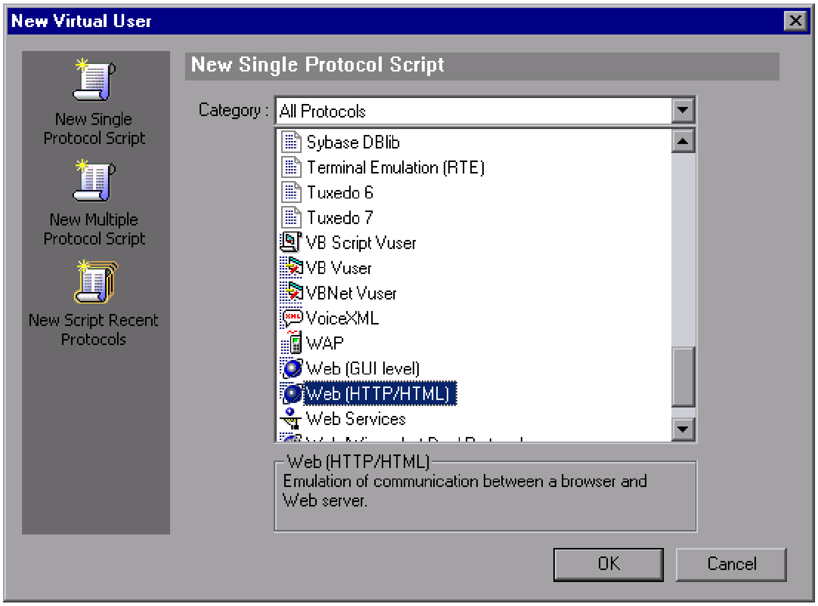
In the Launcher window, click the **Load Testing** tab.

Click **Create/Edit Scripts**. VuGen's Start Page opens.

3 Create a blank Web script.

In the **Scripts** tab in the VuGen Start Page, click **New Vuser Script**.

The New Virtual User dialog box opens, displaying the options for a new single protocol script.



Category : All Protocols

Web (HTTP/HTML)
Emulation of communication between a browser and Web server.

A protocol is the language that your client uses to communicate with the back-end of the system. HP Web Tours is a Web-based application, so you will create a Web Vuser Script.

Make sure the Category type is **All Protocols**. VuGen displays a list of all of the available protocols for a single protocol script. Scroll down the list, select **Web (HTTP/HTML)**, and click **Create** to create an empty Web script.

Note: Advanced users can record several protocols during a single recording session in a *multi-protocol* script. In this tutorial, you create a single protocol script of *Web* type. The procedure for recording other types of single protocols or multi-protocol scripts is similar to the method you will use for recording a Web script.

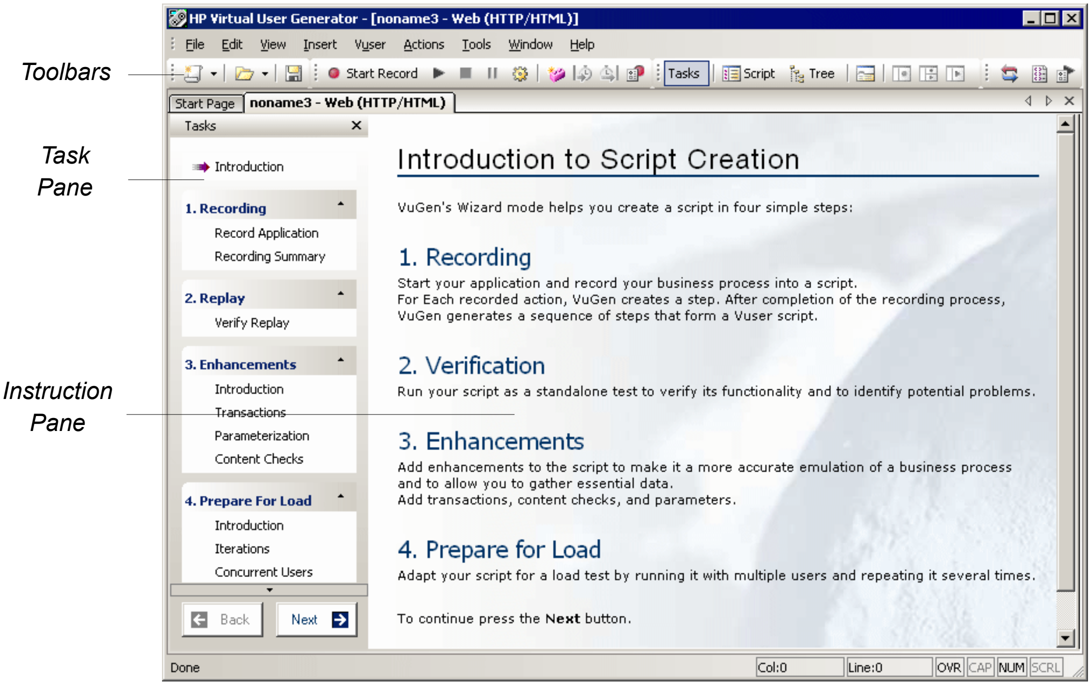
Using VuGen Wizard Mode

Tasks

The empty script opens in VuGen's *wizard* mode with the *Task pane* displayed on the left. (If the task pane is not displayed click the **Tasks** button on the toolbar. If the Start Recording dialog box opens automatically, click **Cancel**).

VuGen's wizard takes you through a step by step process of creating a script and then adapting it for your test environment.

The Task pane lists each step or task in the script creation process. As you proceed through each step, VuGen displays detailed instructions and guidelines in the main area of the window.



Introduction to Script Creation

VuGen's Wizard mode helps you create a script in four simple steps:

1. Recording

Start your application and record your business process into a script.
For Each recorded action, VuGen creates a step. After completion of the recording process, VuGen generates a sequence of steps that form a Vuser script.

2. Verification

Run your script as a standalone test to verify its functionality and to identify potential problems.

3. Enhancements

Add enhancements to the script to make it a more accurate emulation of a business process and to allow you to gather essential data.
Add transactions, content checks, and parameters.

4. Prepare for Load

Adapt your script for a load test by running it with multiple users and repeating it several times.

To continue press the **Next** button.

You can customize your VuGen window to show or hide the various toolbars. To show or hide a toolbar, select **View** > **Toolbars** and toggle the check mark adjacent to the desired toolbar.

You return to the VuGen wizard at any stage by opening the task pane and clicking on one of the task steps.

How do I record a business process to create a script?

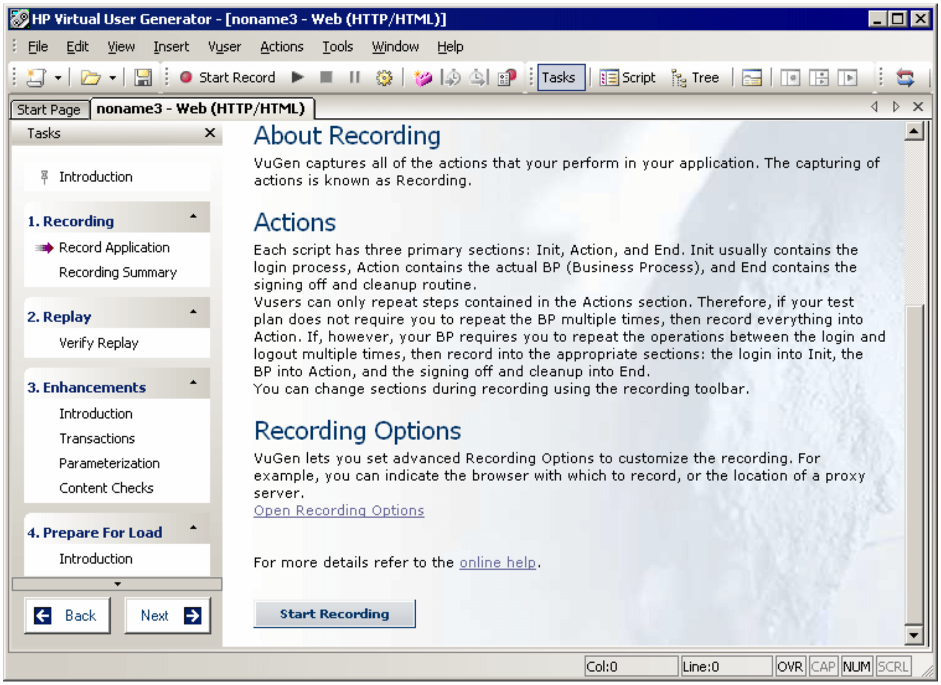
The next step in creating user emulation is recording the events performed by a real user. In the previous section, you created an empty Web script. Now you can begin to record events directly into the script. In this section, you will track the events of one passenger reserving a flight from Denver to Los Angeles and then checking the flight itinerary.

To record the script:

1 Start recording on the HP Web Tours Website.

Click **Record Application** in step 1 in the Task pane.

Click **Start Recording** at the bottom of the instruction pane.



About Recording

VuGen captures all of the actions that your perform in your application. The capturing of actions is known as Recording.

Actions

Each script has three primary sections: Init, Action, and End. Init usually contains the login process, Action contains the actual BP (Business Process), and End contains the signing off and cleanup routine.
Vusers can only repeat steps contained in the Actions section. Therefore, if your test plan does not require you to repeat the BP multiple times, then record everything into Action. If, however, your BP requires you to repeat the operations between the login and logout multiple times, then record into the appropriate sections: the login into Init, the BP into Action, and the signing off and cleanup into End.
You can change sections during recording using the recording toolbar.

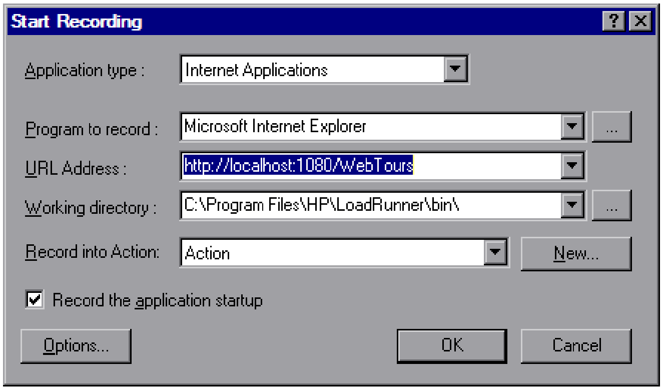
Recording Options

VuGen lets you set advanced Recording Options to customize the recording. For example, you can indicate the browser with which to record, or the location of a proxy server.
Open Recording Options

For more details refer to the online help.

Alternatively you can select **Vuser** > **Start Record** or click the **Start Record** button in the Toolbar at the top of the page.

The Start Recording dialog box opens.



Start Recording

Application type : Internet Applications

Program to record : Microsoft Internet Explorer

URL Address : http://localhost:1080/WebTours

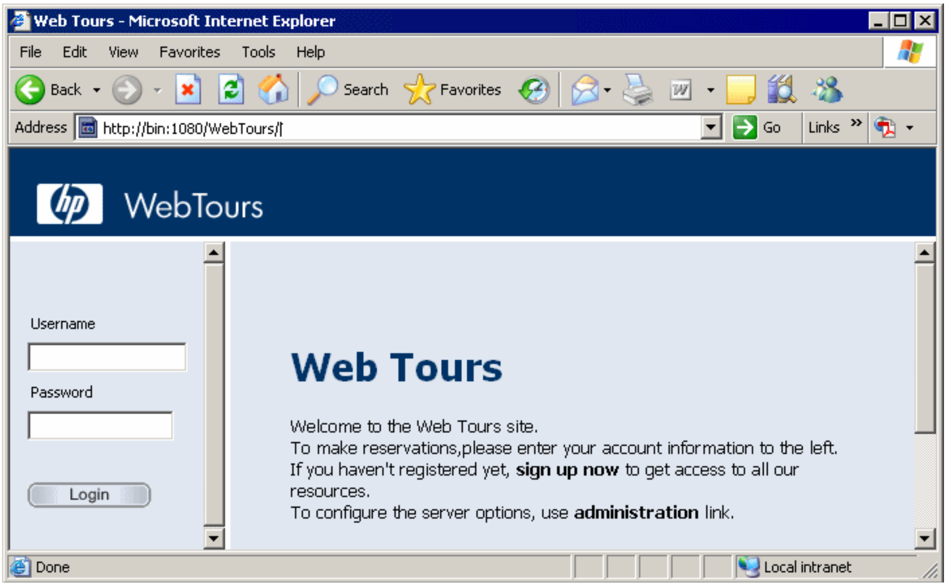
Working directory : C:\Program Files\HP\LoadRunner\bin\

Record into Action: Action

☑ Record the application startup

In the **URL** address box, type http://localhost:1080/WebTours. In the **Record into Action** box, select **Action**. Click **OK**.

A new Web browser opens and displays the HP Web Tours site.



Web Tours - Microsoft Internet Explorer

Address http://bin:1080/WebTours/

WebTours

Username

Password

Web Tours

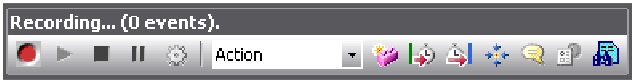
Welcome to the Web Tours site.
To make reservations,please enter your account information to the left.
If you haven't registered yet, **sign up now** to get access to all our resources.
To configure the server options, use **administration** link.

Done

Local intranet

Note: If there is an error opening the site, make sure that the web server is running. To start the server select **Start** > **Programs** > **LoadRunner** > **Samples** > **Web** > **Start Web Server**.

The floating Recording toolbar opens.



2 **Log on to the HP Web Tours Website.**

For the **User Name** enter *jojo* and for the **Password** enter *bean*. Click **Login**. A welcome page opens.

3 **Enter flight details.**

Click **Flights**. The Find Flight page opens:

- **Departure City**: Denver (default)
- **Departure Date:** Keep the default, current date
- **Arrival City:** Los Angeles
- **Return Date:** Keep the default, tomorrow's date
- **Seating Preference**: Aisle

Keep the rest of the default settings and click **Continue**. The Search Results page opens.

4 **Select a flight.**

Click **Continue** to accept the default flight selections. The Payment Details page opens.

5 **Enter payment information and book flight.**

Enter 12345678 in the **Credit Card** box and type 06/10 in the **Exp Date** box. Click **Continue**. The Invoice page opens, displaying your invoice.

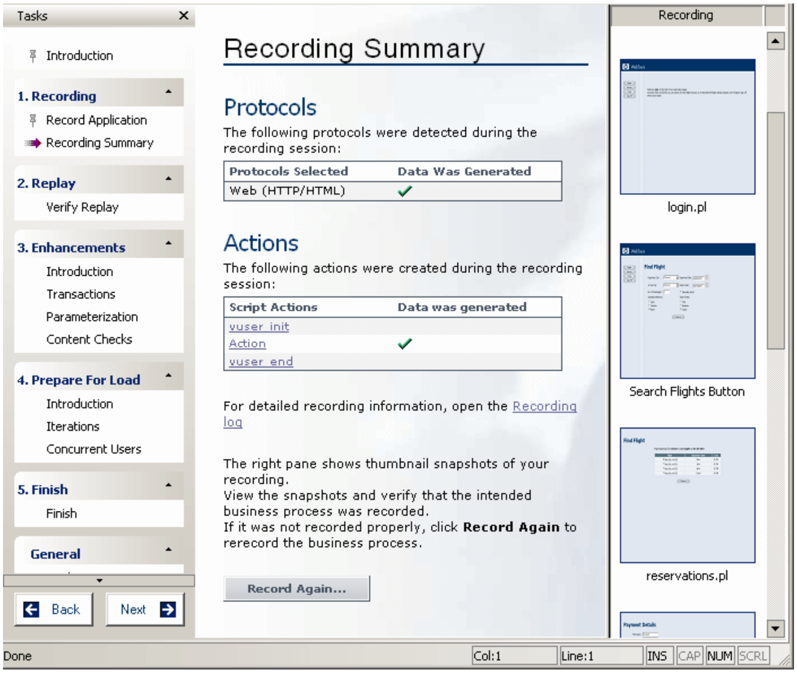
6 Check the itinerary.

Click **Itinerary** in the left pane. The Itinerary page opens.

7 Click **Sign Off** in the left pane.

8 Click **Stop** on the floating toolbar to stop the recording process.

The Code Generation pop up window opens while the Vuser script is generated. The VuGen wizard then automatically proceeds to the next step in the task pane and displays the recording summary. (If you do not see the summary, click **Recording Summary** in the Task pane.)



Recording Summary

Protocols

The following protocols were detected during the recording session:

Protocols Selected	Data Was Generated
Web (HTTP/HTML)	✓

Actions

The following actions were created during the recording session:

Script Actions	Data was generated
vuser_init	
Action	✓
vuser_end	

For detailed recording information, open the Recording log

The right pane shows thumbnail snapshots of your recording.
View the snapshots and verify that the intended business process was recorded.
If it was not recorded properly, click **Record Again** to rerecord the business process.

Record Again...

The recording summary includes the protocol information and a list of the actions created during the session. For each step you performed during recording, VuGen generated a *snapshot*, a picture of the window during recording.

Thumbnails of these recorded snapshots are displayed in the right pane. If, for whatever reason, you want to rerecord your script, you can click the **Record Again** button at the bottom of the page.

9 Select **File** > **Save** or click the **Save** button. Type **basic_tutorial** in the File name box and click **Save**. VuGen saves the file in the LoadRunner script folder and displays the script name in the title bar.

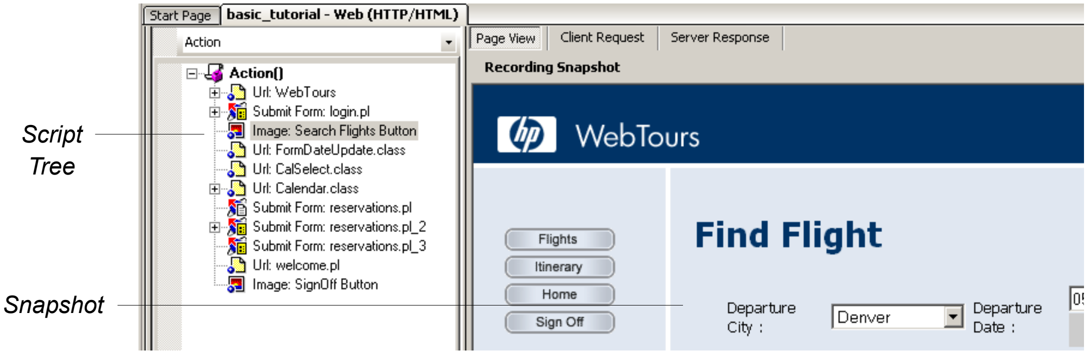
How do I view the script?

You have now recorded a travel agent logging in, booking a flight, and logging off. VuGen recorded your steps from the moment you clicked the **Start Record** button to the moment you clicked the **Stop** button.

You can now view the script inside VuGen. You can view the script in *Tree view* or *Script view*. Tree view is an icon-based view that lists the actions of the Vuser as steps, while Script view is a text-based view that lists the actions of the Vuser as functions.

Tree View

To view the script in Tree view select **View** > **Tree View** or click the Tree view button. To view the Tree View across the whole window, remove the Task pane by clicking the Task button.



Action()
- Url: WebTours
- Submit Form: login.pl
- Image: Search Flights Button
- Url: FormDateUpdate.class
- Url: CalSelect.class
- Url: Calendar.class
- Submit Form: reservations.pl
- Submit Form: reservations.pl_2
- Submit Form: reservations.pl_3
- Url: welcome.pl
- Image: SignOff Button

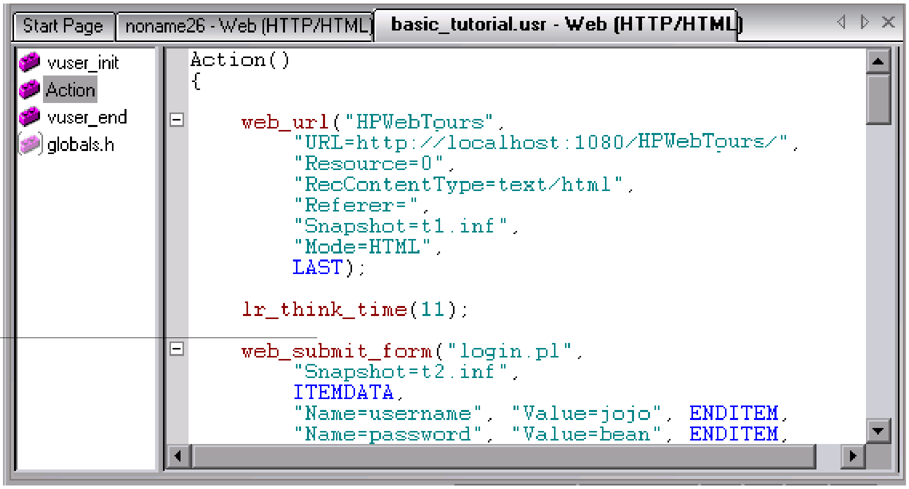
For each step you performed during recording, VuGen generated an icon and a title in the test tree. In Tree view, you see the actions of the user as script steps. Most steps are accompanied by a corresponding snapshot of the recording.

The snapshots make the scripts easier to understand, and easier to share between engineers because you can see exactly which screen was recorded during the recording process. You can compare the snapshots afterwards to verify your script's accuracy. VuGen also creates snapshots of each step during replay.

Click the plus (+) sign adjacent to any of the steps in the test tree. You now see the *Think Time* that was recorded while you were booking a flight. Think Time represents the actual time you waited between steps, and can be used to emulate fast and slow user behavior under load. Think time is a mechanism whereby you can make your load test more accurately reflect a real user's behavior.

Script View

Script view is a text-based view that lists the actions of the Vuser as API functions. To view the script in Script view select **View** > **Script View** or click the **Script View** button.


Start Page
noname26 - Web (HTTP/HTML)
basic_tutorial.usr - Web (HTTP/HTML)
vuser_init
Action
vuser_end
globals.h

```
Action()
{

	web_url("HPWebTours",
		"URL=http://localhost:1080/HPWebTours/",
		"Resource=0",
		"RecContentType=text/html",
		"Referer=",
		"Snapshot=t1.inf",
		"Mode=HTML",
		LAST);

	lr_think_time(11);

	web_submit_form("login.pl",
		"Snapshot=t2.inf",
		ITEMDATA,
		"Name=username", "Value=jojo", ENDITEM,
		"Name=password", "Value=bean", ENDITEM,
```

VuGen Editor

In Script view, VuGen shows the script in an editor with color coded functions and their argument values. You can type C or LoadRunner API functions, as well as control flow statements, directly into this window.

Note: LoadRunner uses ANSI C. As your scripts become more advanced, you can leverage C to extend LoadRunner beyond record and playback.

Where To Go From Here

Now that you are familiar with recording and viewing a basic script, you can proceed to Lesson 4, “Playing Back Your Script.”

4

Playing Back Your Script

By recording a set of typical user actions such as booking a flight, you created a real user emulation. You replay your recorded script to verify that it runs properly before you incorporate it into a load testing scenario. During replay, you can view the actions in a browser and see if everything is as you expect it to be. If the script doesn't replay properly, you may need to add correlations, as described in Lesson 5, "Solving Common Playback Problems."

Before replaying the script, you can configure run-time settings, which help you set the behavior of the Vuser.

In this lesson you will cover the following topics:

- How do I set the run-time behavior?
- How do I watch my script running in real time?
- Where can I view information about the replay?
- How do I know if my test passed?
- How do I search in or filter the results?

How do I set the run-time behavior?

LoadRunner Run-Time settings let you emulate different kinds of real user activity and behavior. For example, you could emulate a user who responds immediately to output from the server, or a user who stops and thinks before each response. You can also configure run-time settings to specify how many times the Vuser should repeat a set of actions and how often.

There are general run-time settings, and settings that are specific to certain Vuser types. For example, for a Web emulation, you could instruct your Vusers to replay your script in Netscape instead of Internet Explorer. Specific settings will be covered in Lesson 6, "Preparing a Script for Load Testing."

In this lesson, general run-time settings that apply to all types of scripts will be discussed. They include:

- **Run Logic.** The number of repetitions
- **Pacing.** The time to wait between repetitions
- **Think Time.** The time the user stops to think between steps
- **Log.** The level of information that you want to gather during playback

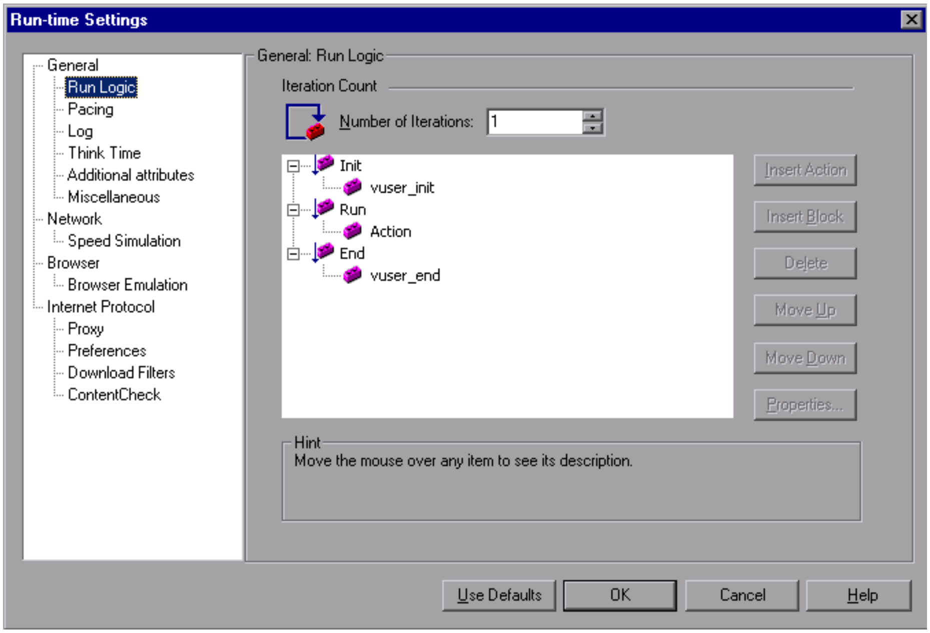
Note that you can also modify the run-time settings from the LoadRunner Controller. This will be discussed in a later lesson.

1 Open the run-time settings.

Ensure that the Task pane is displayed (If not click the **Task** button). Click **Verify Replay** in the task pane.

Under the heading **Run-Time Settings** in the instruction pane click the **Open Run-Time Settings** hyperlink.

Alternatively you can press F4 or click the **Run-Time Settings** button in the toolbar. The Run-Time Settings dialog box opens.

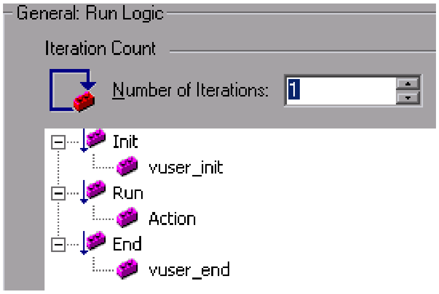


Number of Iterations: 1

Move the mouse over any item to see its description.

2 Open the Run Logic settings.

Select the **Run Logic** node.

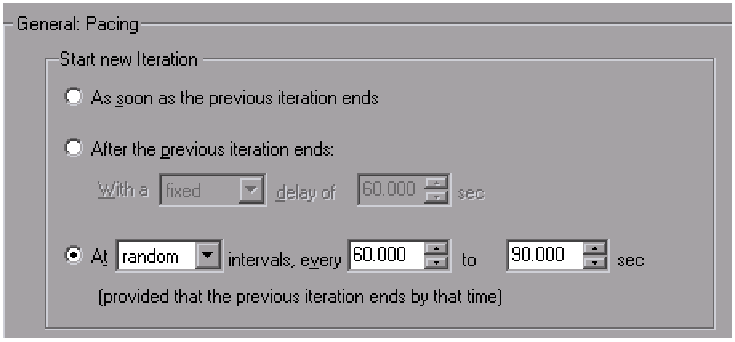


Number of Iterations: 1

In this node you set the number of *iterations*, or the number of times to repeat the activity in succession. Set the number of iterations to 2.

3 Set the Pacing settings.

Select the **Pacing** node.



General: Pacing

Start new Iteration

○ As soon as the previous iteration ends

○ After the previous iteration ends:

With a fixed delay of 60.000 sec

● At random intervals, every 60.000 to 90.000 sec

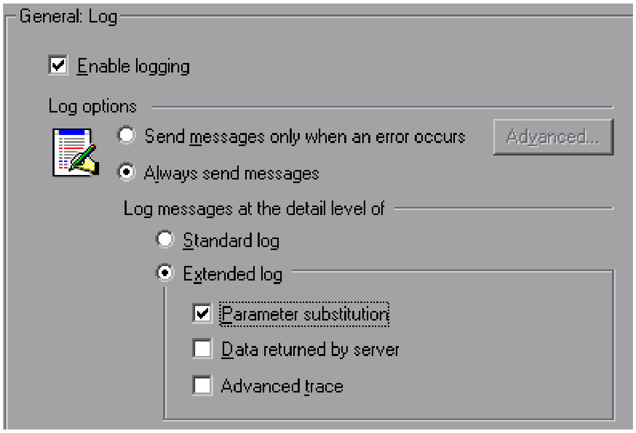
(provided that the previous iteration ends by that time)

This node lets you control the time between iterations. You will specify a random time. This accurately emulates a real-life setting where the user waits between actions, but at random intervals—you do not see real users waiting exactly 60 seconds between repetitions.

Select the third option and select the following:
At **random** intervals, every **60.00** to **90.00** seconds.

4 Set the Log settings.

Select the **Log** node.



General: Log

☑ Enable logging

Log options

○ Send messages only when an error occurs

● Always send messages

Log messages at the detail level of

○ Standard log

● Extended log

☑ Parameter substitution

□ Data returned by server

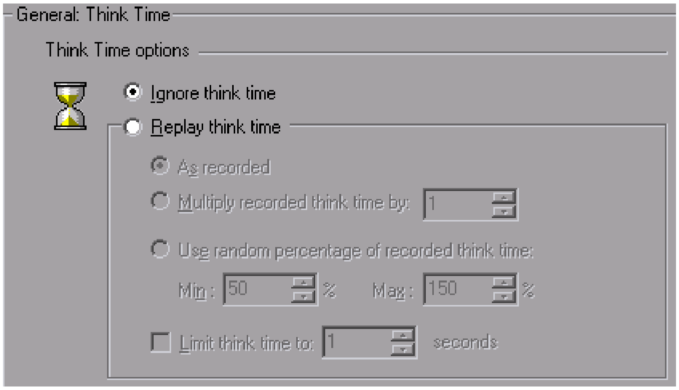
□ Advanced trace

The Log settings indicate how much information to log while running the test. During development, you may select to enable some logging for debugging purposes, but once you verify that your script is functional, you can enable logging for errors only or disable it.

Select **Extended log** and enable **Parameter substitution**. This option will be relevant for the following lesson at which point it will be discussed.

5 View the Think Time settings.

Select the **Think Time** node.



General: Think Time

Think Time options

● Ignore think time

○ Replay think time

● As recorded

○ Multiply recorded think time by: 1

○ Use random percentage of recorded think time:

Min : 50 % Max : 150 %

□ Limit think time to: 1 seconds

Do not make any changes. You will set the think time from the Controller. Keep in mind that when you run the script in VuGen, it will run quickly since it will not include think time.

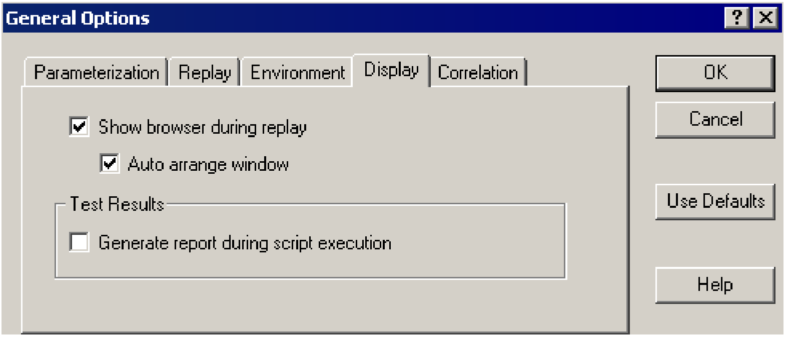
6 Click **OK** to close the Run-Time Settings dialog box.

How do I watch my script running in real time?

VuGen's *run-time viewer* feature displays the Vuser activities in real time as you playback the recorded script.

By default, VuGen runs your test in the background, without displaying an animation of the actions in your script. For this tutorial, however, you will instruct VuGen to display the actions in a viewer that lets you see how VuGen executes each step. The viewer is not an actual browser—it only displays snapshots of the pages that are returned to the Vuser.

1. Select **Tools** > **General Options,** and select the **Display** tab.
2. Select **show browser during replay** and **Auto arrange window** options.



☑ Show browser during replay

☑ Auto arrange window

Test Results

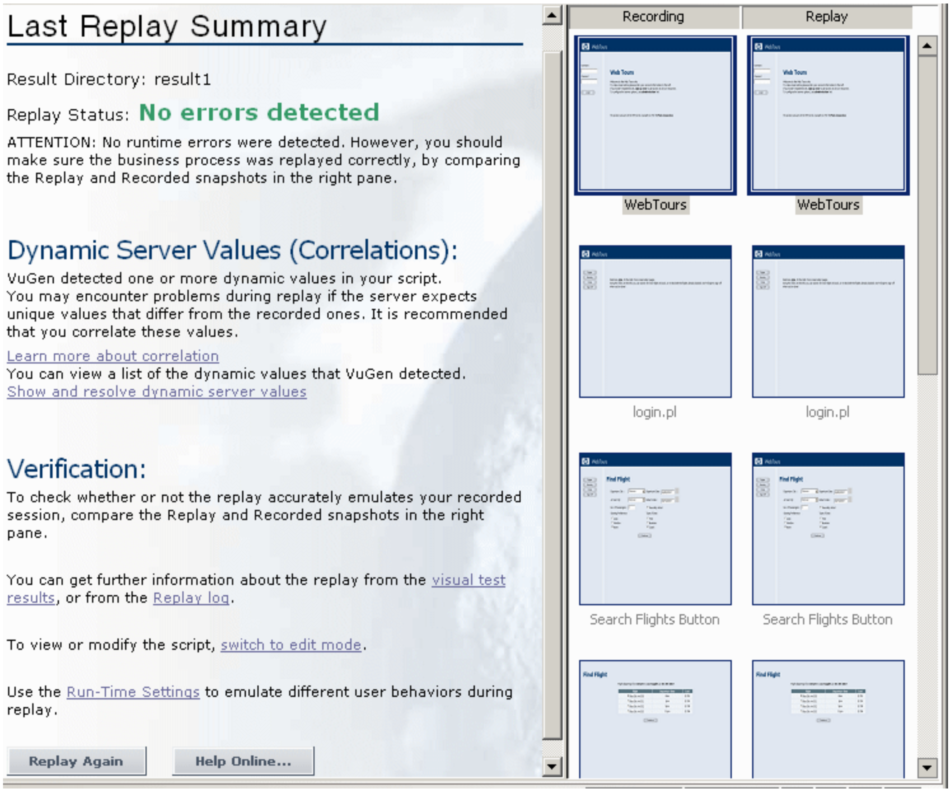
□ Generate report during script execution

3. Click **OK** to close the General Options dialog box.
4. Click **Verify Replay** in the Task Pane and then click the **Start Replay** button at the bottom of the instruction pane. Alternatively you can press F5 or click the **Run** button in the toolbar.
5. If the Select Results Directory dialog box opens and asks you where you want to store the results directory, accept the default name and click **OK.**

   After a few moments, VuGen opens a run-time viewer and begins running the script in either script view or tree view, depending on what you last had open. In the run-time viewer, you visually observe the Vuser's actions. Note how the replay is an exact playback of the steps you performed during recording.

Where can I view information about the replay?

When the script stops running, you can view a summary of the replay in the wizard. To view the Last Replay Summary, click **Verify Replay**.



Last Replay Summary

Result Directory: result1

Replay Status: No errors detected

ATTENTION: No runtime errors were detected. However, you should make sure the business process was replayed correctly, by comparing the Replay and Recorded snapshots in the right pane.

Dynamic Server Values (Correlations):

VuGen detected one or more dynamic values in your script. You may encounter problems during replay if the server expects unique values that differ from the recorded ones. It is recommended that you correlate these values.

Learn more about correlation

You can view a list of the dynamic values that VuGen detected.

Show and resolve dynamic server values

Verification:

To check whether or not the replay accurately emulates your recorded session, compare the Replay and Recorded snapshots in the right pane.

You can get further information about the replay from the visual test results, or from the Replay log.

To view or modify the script, switch to edit mode.

Use the Run-Time Settings to emulate different user behaviors during replay.

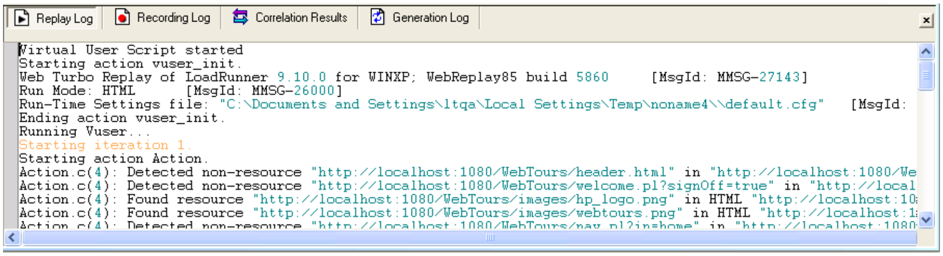
The **Last Replay Summary** lists any errors that may have been detected and displays thumbnails of the Recording and Replay snapshots. You can compare snapshots and look for discrepancies between what you recorded and the replay.

You can also look at the Vuser activity by reviewing a textual summary of the events. VuGen's Replay Log tab in the Output window shows this color-coded information.

In this section, you will open the replay log and locate some of the events and notifications.

To view the Replay Log:

1 Click the **Replay Log** hyperlink in the instruction pane. Alternatively click the **Show/Hide output** button in the toolbar or select **View** > **Output Window** from the menu. Then click the **Replay Log** tab.



```
Virtual User Script started
Starting action vuser_init.
Web Turbo Replay of LoadRunner 9.10.0 for WINXP; WebReplay85 build 5860     [MsgId: MMSG-27143]
Run Mode: HTML      [MsgId: MMSG-26000]
Run-Time Settings file: "C:\Documents and Settings\ltqa\Local Settings\Temp\noname4\\default.cfg"   [MsgId:
Ending action vuser_init.
Running Vuser...
Starting iteration 1.
Starting action Action.
Action.c(4): Detected non-resource "http://localhost:1080/WebTours/header.html" in "http://localhost:1080/We
Action.c(4): Detected non-resource "http://localhost:1080/WebTours/welcome.pl?signOff=true" in "http://local
Action.c(4): Found resource "http://localhost:1080/WebTours/images/hp_logo.png" in HTML "http://localhost:10
Action.c(4): Found resource "http://localhost:1080/WebTours/images/webtours.png" in HTML "http://localhost:1
Action.c(4): Detected non-resource "http://localhost:1080/WebTours/nav.pl?in=home" in "http://localhost:1080
```

2 Press Ctrl+F in the Replay Log to open the Find dialog box. Locate the following items:

- **Started**, **Terminated**. The beginning and end of the script run—Virtual User Script Started, Vuser Terminated.
- **iteration.** The beginning and end of the iteration and the iteration number (orange lettering).

VuGen displays successful steps in green and errors in red. For example, if your connection broke in the middle of the test, VuGen indicates the line number of the error and displays the whole line in red text.

3 Double-click a line in the Replay Log. VuGen takes you to the corresponding step in the script and highlights the step in script view.

How do I know if my test passed?

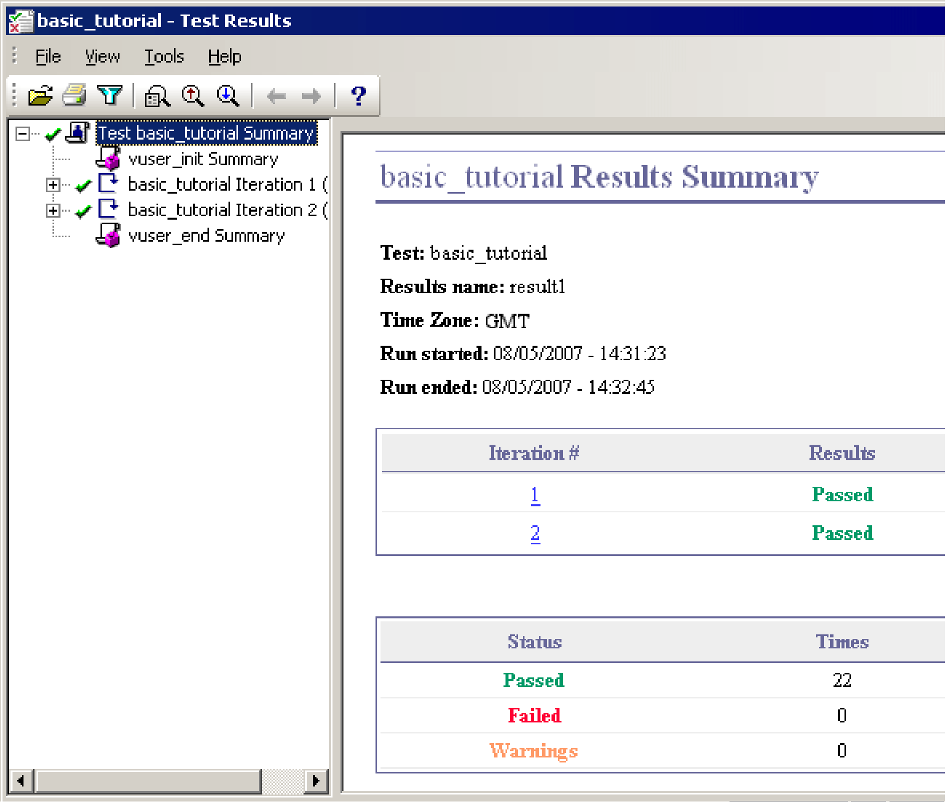
After you playback the events that you recorded, you need to look at the results and see if everything succeeded. If something failed, you want to know when and why it failed.

In this section you will view and analyze the results of your script run. VuGen summarizes the results of the replay in the Test Results window.

To view test results:

1. To return to the wizard click **Verify Replay** in the Task Pane.

2. In the instruction pane under the heading **Verification**, click the **visual test results** hyperlink. Alternatively, select **View** > **Test Results.** A new results window opens.


basic_tutorial - Test Results
File
View
Tools
Help
Test basic_tutorial Summary
vuser_init Summary
basic_tutorial Iteration 1 (
basic_tutorial Iteration 2 (
vuser_end Summary

basic_tutorial Results Summary

Test: basic_tutorial

Results name: result1

Time Zone: GMT

Run started: 08/05/2007 - 14:31:23

Run ended: 08/05/2007 - 14:32:45

Iteration #	Results
1	Passed
2	Passed

Status	Times
Passed	22
Failed	0
Warnings	0

When the Test Results window first opens, it contains two panes: Tree pane (on the left) and Summary pane (on the right).

The Tree pane contains the results tree. Each iteration is numbered. The Summary pane contains the details of the test.

The top table indicates which iterations passed and which failed. The test is considered to have passed when VuGen's Vuser successfully navigates through the HP Web Tours site according to the original recording.

The bottom table indicates whether transactions and checkpoints passed or failed. You will add these features to your test later on in the tutorial.

In the next section, you will drill down into the test results to determine if the script reached the intended Web pages during replay.

How do I search in or filter the results?

If your test results indicate that something failed, you can drill down and locate the point of failure.

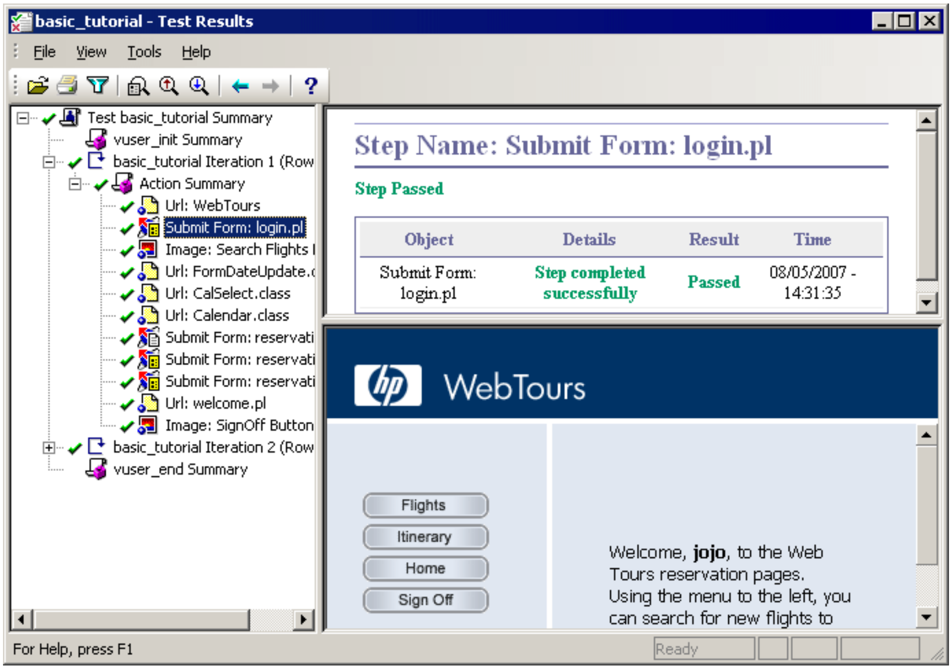
In the Test Results window, you can expand the test tree and view the results of each step separately. The Test Results window shows a snapshot of the replay during that iteration.

1 Expand an iteration branch.

Expand the branch **Iteration 1** and then expand the **Action Summary** branch in the left pane by clicking the plus sign. The expanded branch shows a list of the steps performed in that iteration.

2 Show a result snapshot.

Select the step, **Submit Form**. The Test Results window displays the replay snapshot associated with that step.



Step Name: Submit Form: login.pl

Step Passed

Object	Details	Result	Time
Submit Form: login.pl	Step completed successfully	Passed	08/05/2007 - 14:31:35

Welcome, **jojo**, to the Web Tours reservation pages.
Using the menu to the left, you can search for new flights to

3 View the step summary.

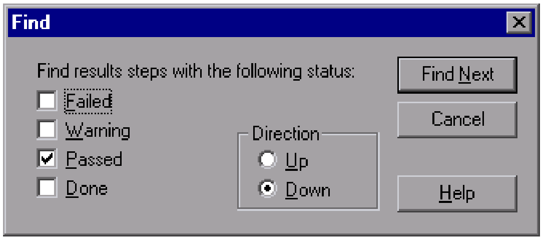
The top right pane of the Test Results window shows the step summary information: the object or step name, the details about whether the page loaded successfully, the result (**Passed**, **Failed**, **Done**, or **Warning**), and the time that the step was performed.

4 Search for a Result status.

You can search the test results for the words **Passed** or **Failed**.

This is helpful, as in a case where the overall Results Summary indicates that the test failed, you can determine where it failed.

To Search the test results, select **Tools** > **Find** or click the **Find** button. The Find dialog box opens.



Find

Find results steps with the following status:

☐ Failed
☐ Warning
☑ Passed
☐ Done

Direction
○ Up
● Down

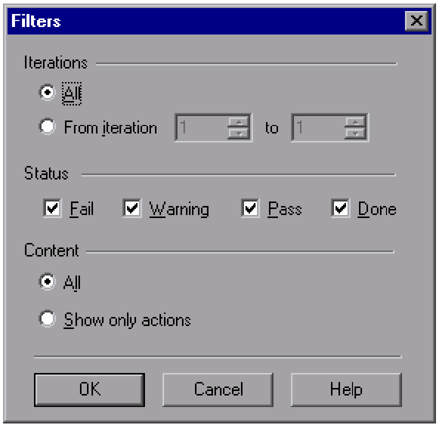
Select the **Passed** option, make sure no other options are selected, and click **Find Next**. The Results window shows the first step, where the status was **Passed**.

Once again, select **Tools** > **Find** or click the **Find** button. In the Find dialog box, select the **Failed** option, clear the **Passed** option, and click **Find Next**. The Results window cannot find any failed results.

5 Filter the results.

You can filter the Test Results window to display a specific iteration or status. For example, you can filter it to show only **Failed** status.

To filter the results select **View** > **Filters** or click the **Filters** button. The Filters dialog box opens.



Iterations

● All

○ From iteration 1 to 1

Status

☑ Fail ☑ Warning ☑ Pass ☑ Done

Content

● All

○ Show only actions

In the **Status** section, select **Fail** and clear all the other options. In the **Content** section, select **All** and click **OK**. The left pane becomes empty, since there were no failures.

6 Close the Test Results window.

Select **File** > **Exit**.

Where To Go From Here

You have now successfully run your test of the HP Web Tours application. You can proceed to Lesson 5, “Solving Common Playback Problems.”

5

Solving Common Playback Problems

After you create a script, you validate it by running it from VuGen. Occasionally, a simple playback will fail, even though the recording of the same actions succeeded.

Many applications use dynamic values that change each time you use the application. For example, some servers assign a unique session ID for every new session. When you try to replay a recorded session, the application creates a new session ID that differs from the recorded session ID.

LoadRunner addresses this issue through *correlation*. Correlation saves the changing values, in our case the session ID, to a parameter. When running the emulation, the Vuser does not use the recorded value—instead, it uses the new session ID, assigned to it by the server.

In this lesson you will observe how LoadRunner automatically solves the issue of dynamic values.

In this lesson you will cover the following topics:

- ➤ Preparing HP Web Tours for Playback Errors
- ➤ How do I work with unique server values?

Preparing HP Web Tours for Playback Errors

To illustrate a common playback failure, you need to modify a setting in the HP Web Tours application. This setting tells the HP Web Tours Web Server not to allow duplicate session IDs.

1 **Open HP Web Tours.**

Select **Start** > **Programs** > **LoadRunner** > **Samples** > **Web** > **HP Web Tours Application**. A browser opens with the HP Web Tours home page.

2 **Change the server options.**

Click the **administration** link on the HP Web Tours home page. The administration page opens.

Select the 3rd checkbox entitled: **Set LOGIN form's action tag to an error page.** Scroll down to the bottom of the page and click **Update**.

Scroll down to the bottom of the page and click the **Return to the Web Tours Homepage** link.

This setting tells the server not to allow duplicate session IDs.

3 **Close the browser.**

How do I work with unique server values?

In the modified configuration of HP Web Tours, the server assigns a unique session ID to the Vuser. If you try to play back the script, it will fail.

To overcome this issue, you use VuGen to automatically detect the need to correlate the session ID. After you run the script, VuGen prompts you to scan the script for correlations.

You will instruct VuGen to insert a step that saves the original session ID to a parameter. In each replay session, VuGen saves the new unique session ID to a parameter. In the subsequent steps, it uses the saved value instead of the originally recorded value.

1 Record a new script with dynamic values.

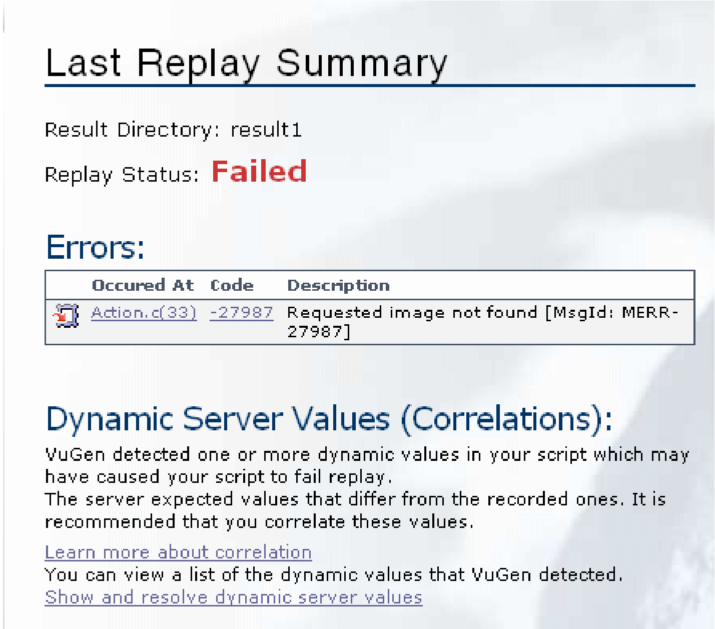
Record a new script with the same steps that you recorded previously in Lesson 3, “How do I record a business process to create a script?” on page 31 and save the script as Basic_Tutorial_Cor.

2 Replay the script.

Click **Verify Replay** in the Task Pane and click the **Start Replay** button at the bottom of the instruction pane. VuGen runs the new script. You may notice several error messages in the **Replay Log** tab in the Output window, indicated by the red-colored text.

3 View the Replay Summary

Click Verify Replay in the Task Pane to view the Last Replay Summary.



Last Replay Summary

Result Directory: result1

Replay Status: Failed

Errors:

	Occured At	Code	Description
	Action.c(33)	-27987	Requested image not found [MsgId: MERR-27987]

Dynamic Server Values (Correlations):

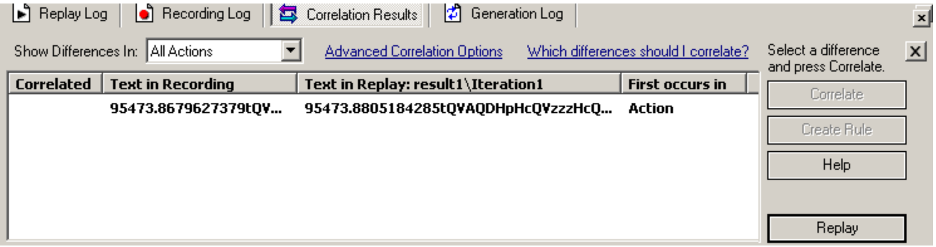
VuGen detected one or more dynamic values in your script which may have caused your script to fail replay.
The server expected values that differ from the recorded ones. It is recommended that you correlate these values.

Learn more about correlation
You can view a list of the dynamic values that VuGen detected.
Show and resolve dynamic server values

The summary above shows that your script failed.

4 Scan the script for correlations.

Click the link **Show and resolve dynamic server values** in the instruction pane under the heading Dynamic Server Values. VuGen scans the script, searching for differences between the recorded values and the playback values. VuGen displays a list of these differences that may require correlation in the Correlation Results tab in the Output window.

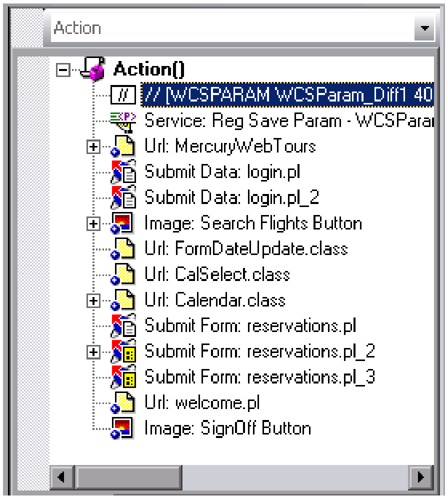

Replay Log
Recording Log
Correlation Results
Generation Log
Advanced Correlation Options
Which differences should I correlate?
Select a difference and press Correlate.
Correlate
Create Rule
Help
Replay

Show Differences In: All Actions

Correlated	Text in Recording	Text in Replay: result1\Iteration1	First occurs in
	95473.8679627379tQV...	95473.8805184285tQVAQDHpHcQVzzzHcQ...	Action

5 Correlate the Session ID.

Select the first entry in the Correlation Results tab, and click **Correlate**. VuGen inserts a new step at the top of the script saving the original session ID to a parameter. In each replay session, VuGen saves the new unique session ID to a parameter. In the subsequent steps, it uses the saved value instead of the originally recorded value. Select **View** > **Tree View** to view this new step in the test tree.



Action

Action()
- // [WCSPARAM WCSParam_Diff1 40
- Service: Reg Save Param - WCSPara
- Url: MercuryWebTours
- Submit Data: login.pl
- Submit Data: login.pl_2
- Image: Search Flights Button
- Url: FormDateUpdate.class
- Url: CalSelect.class
- Url: Calendar.class
- Submit Form: reservations.pl
- Submit Form: reservations.pl_2
- Submit Form: reservations.pl_3
- Url: welcome.pl
- Image: SignOff Button

6 Examine the syntax of the correlation statement.

View > **Script View** to view the correlation statement in the script. The statement that VuGen added to the script looks like this:

```
web_reg_save_param (“WCSParam_Diff1”,
        “LB=userSession value=”,
        “RB=>”,
        “Ord=1”,
        “RelFrameId=1.2.1”,
        “Search=Body”,
        LAST);
```

This statement means “Check the server response for data located between the following two strings:”

➤ Left boundary: **userSession value=**

➤ Right boundary: >

The statement instructs VuGen to save the first occurrence of this data to a parameter called **WCSParam_Diff1**.

1 Play the script again.

Select **Vuser** > **Run** to replay the script again. When the replay ends, select **View** > **Tree View.** Look in the Replay Log tab. Note that VuGen no longer issues the red-colored error messages.

In the script, right-click **Service: Reg Save Param** and select **Go to step in Replay Log**. VuGen places the cursor at the corresponding line in the replay log. The log indicates that function web_reg_save_param succeeded, indicating that the correlation worked.

2 Fix the server configuration.

Reset the server to ignore unique session IDS.

Select **Start** > **Programs** > **LoadRunner** > **Samples** > **Web** > **HP Web Tours Application** to open HP Web Tours. On the HP Web Tours home page, click the **administration** link. In the administration page, clear the 3rd checkbox entitled: **Set LOGIN form's action tag to an error page.** Scroll down to the bottom of the page and click **Update**. Close the browser.

Automatic Correlation

In this lesson, you scanned a script for correlations **after** recording the user actions.

VuGen also provides a set of configurable correlation rules to handle dynamic values **during** the recording session. For details on automatic correlation, refer to the *HP Virtual User Generator User's Guide*.

Most servers have clear syntax rules, or **contexts**, that they use when creating links and referrals. If you are recording a session with a supported application server, you can use VuGen's built-in correlation rules, and VuGen will detect and correlate the dynamic values during the recording stage.

Where To Go From Here

Now that you are familiar with some of the common playback problems, you can proceed to Lesson 6, "Preparing a Script for Load Testing."

6

Preparing a Script for Load Testing

In the previous lessons, you verified that your script was an accurate emulation of your application. You watched the playback in real time and you verified that the Vuser performed a typical business process.

This, however, is true only for a single user emulation. Will the application work with many users working simultaneously? If so, will the application slow down to an unacceptable level?

The next step is to prepare the script for a load test and set it up to gather response time data. In this lesson you will learn about different methods to enhance the script and to make it more effective for the load testing process.

In this lesson you will cover the following topics:

- How do I measure business processes?
- How do I emulate multiple users?
- How do I verify Web page content?
- How can I produce debugging information?
- Did my test succeed?

How do I measure business processes?

When preparing an application for deployment, you need to measure the duration of specific business processes—how long does it take to log on, book a flight, and so on. These business processes are normally made up of one or more steps or actions within your script. In LoadRunner, you designate a series of actions you want to measure by marking them as transactions.

LoadRunner gathers information about the time it takes to perform a transaction and displays the results in color-coded graphs and reports. You use this information to see if the application meets the original requirements.

You can manually insert a transaction anywhere in your script. The way to mark a user step as a transaction is to place a **start transaction** marker before the first step of the transaction and an **end transaction marker** after the last step.

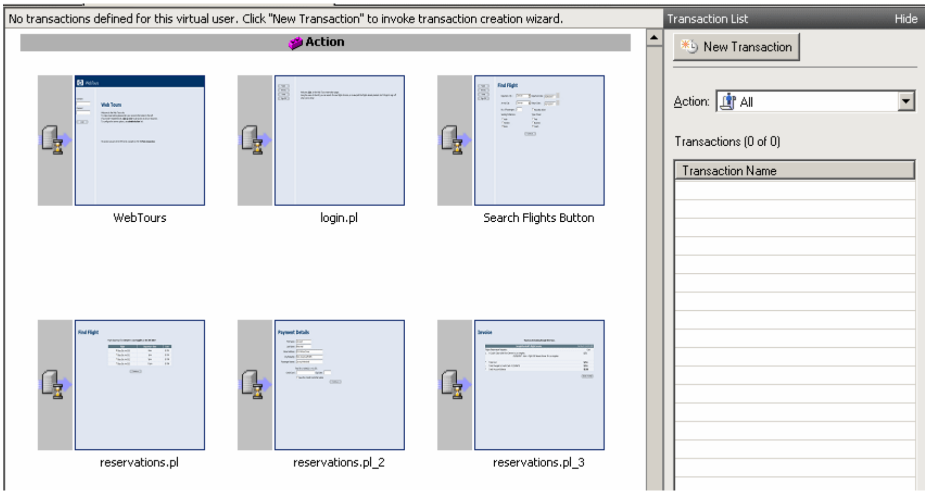
In this section you will insert a transaction in your script to measure the amount of time it takes for the user to find and confirm a flight.

Open the script **Basic_Tutorial** which you created in Lesson 3. If it is still open, you can select the tab displaying its name. Otherwise you can open it from the File menu.

To insert a transaction:

1 Open the Transaction Creation wizard.

Ensure that the Task Pane is displayed. If not click the **Task** button. In the Task Pane under **Enhancements,** click **Transactions**. The Transaction Creation wizard opens.



No transactions defined for this virtual user. Click "New Transaction" to invoke transaction creation wizard.

The Transaction Creation wizard displays thumbnails of the different steps in your script.

Click the **New Transaction** button. You are now able to drag the transaction markers and place them at their designated points in the script. The wizard prompts you to insert a starting point for the transaction.

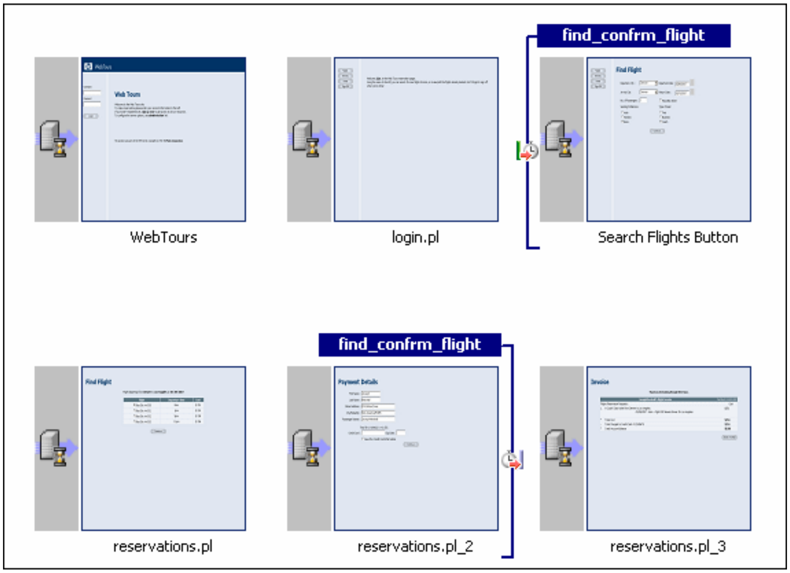
2 Insert a start and end transaction marker.

Using your mouse, position the transaction opening bracket before the third thumbnail entitled **Search flights button** and click to set it in place. The wizard now prompts you to insert an end point.

Using your mouse, position the transaction closing bracket after the fifth thumbnail entitled **reservations.pl_2** and click to set it in place.

3 Specify a name for the transaction.

The wizard prompts you to enter a name for the transaction. Type find_confirm_flight and press ENTER.

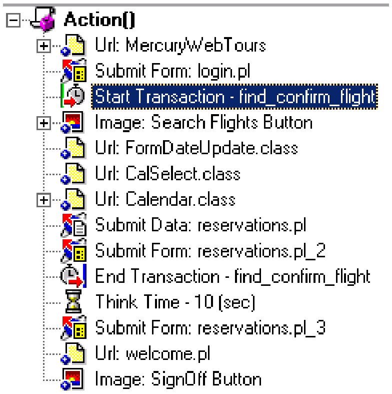


You have just created a new transaction. You can adjust the starting point or end point of the transaction by dragging the markers to different points in the script. You can also rename the transaction by clicking the existing name above the start transaction marker and typing a new one.

4 Observe the transaction in Tree View.

Go to Tree view by choosing **View** > **Tree View** or by clicking the Tree view button in the toolbar.

Notice how the **Start Transaction** and **End Transaction** markers have now been added to the tree as new steps at the precise points where you inserted them.



```
Action()
  Url: MercuryWebTours
  Submit Form: login.pl
  Start Transaction - find_confirm_flight
  Image: Search Flights Button
  Url: FormDateUpdate.class
  Url: CalSelect.class
  Url: Calendar.class
  Submit Data: reservations.pl
  Submit Form: reservations.pl_2
  End Transaction - find_confirm_flight
  Think Time - 10 (sec)
  Submit Form: reservations.pl_3
  Url: welcome.pl
  Image: SignOff Button
```

How do I emulate multiple users?

In your emulation, you tracked a user booking a flight and choosing an aisle seat. In a real-life setting, however, different users will have varying preferences. To improve your test, you need to check if the booking will work when users select different seating preferences (aisle, window, or none).

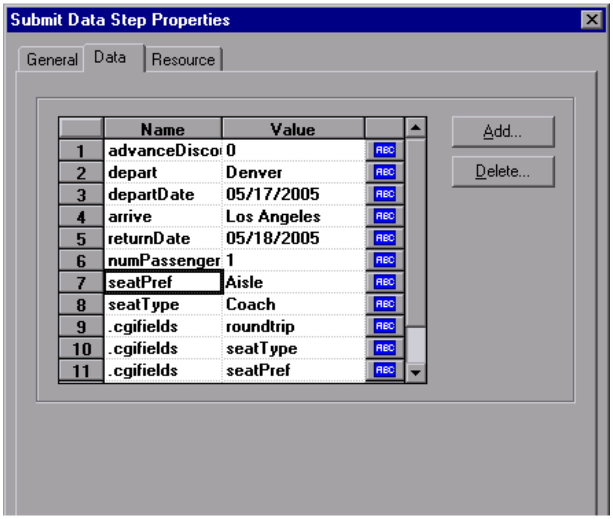
To accomplish this, you will **parameterize** the script. This means that you take the recorded value, **Aisle**, and replace it with a parameter. You will place values for the parameter in a parameter file. When you run the script, the Vuser will take values from the parameter file (aisle, window, or none) thereby emulating a true travel agency environment.

To parameterize your script:

1 Find the section where you want to vary the data.

Go to Tree view by selecting **View** > **Tree View.**

In the Test Tree, double-click the **Submit Data: reservations.pl** step. The Submit Data Step Properties dialog box opens.



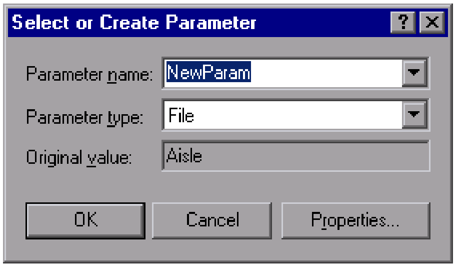
	Name	Value	
1	advanceDisco	0	ABC
2	depart	Denver	ABC
3	departDate	05/17/2005	ABC
4	arrive	Los Angeles	ABC
5	returnDate	05/18/2005	ABC
6	numPassenger	1	ABC
7	seatPref	Aisle	ABC
8	seatType	Coach	ABC
9	.cgifields	roundtrip	ABC
10	.cgifields	seatType	ABC
11	.cgifields	seatPref	ABC

The **ABC** icons in the right column indicate that the arguments are constants.

2 Indicate that the fixed value will be a varying value.

Select **seatPref** in the seventh line.

Click the **ABC** icon adjacent to **Aisle**. The Select or Create Parameter dialog box opens.



Parameter name: NewParam

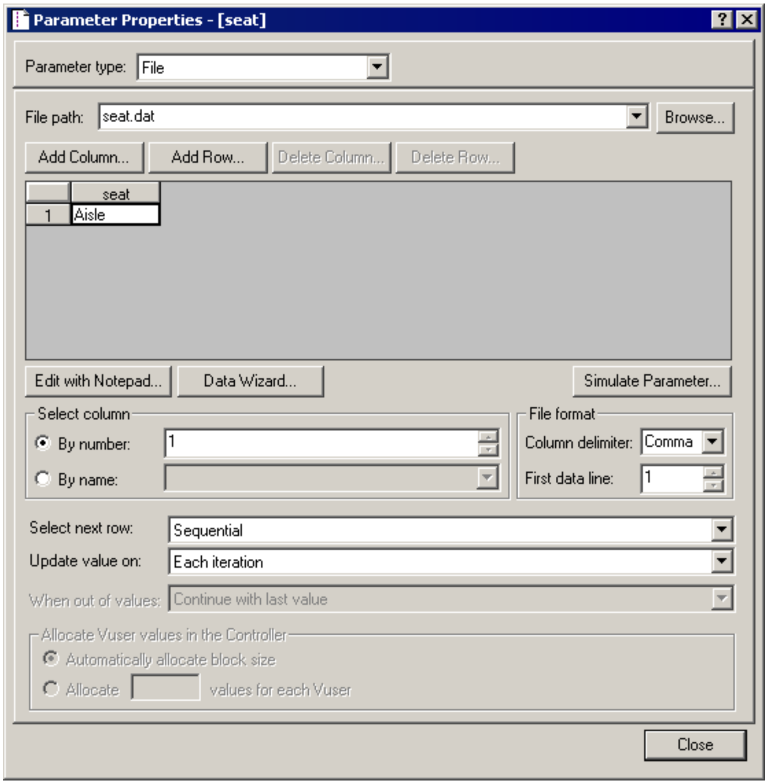
Parameter type: File

Original value: Aisle

3 Create a parameter.

Specify a parameter name, seat, using the **File** parameter type. Click **OK**. VuGen replaces the **ABC** icon with a parameter icon.

4 Click the parameter icon adjacent to **{seat}** and select **Parameter Properties**. The Parameter Properties dialog box opens.



Parameter Properties - [seat]

Parameter type: File

File path: seat.dat

	seat
1	Aisle

Select column

● By number: 1

○ By name:

File format

Column delimiter: Comma

First data line: 1

Select next row: Sequential

Update value on: Each iteration

When out of values: Continue with last value

Allocate Vuser values in the Controller

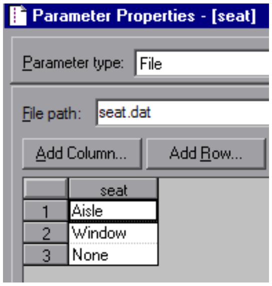
● Automatically allocate block size

○ Allocate values for each Vuser

5 Specify some sample values to vary the data.

Click **Add Row.** VuGen adds a row to the table. Replace the word Value with Window.

Click **Add Row.** VuGen adds a row to the table. Replace the word Value with None.



Parameter type: File

File path: seat.dat

	seat
1	Aisle
2	Window
3	None

Note: The values are not case sensitive.

Keep the default settings in the **Select column** and **File format** sections of the dialog box.

6 Define how the test will vary the data.

Keep the default setting that instructs VuGen to take sequential values for each iteration—not random values.

- **Select next row:** Sequential
- **Update value on**: Each iteration

7 Click **Close** to close the Parameter Properties dialog box, and click **OK** to close the Step Properties dialog box.

You have now created a parameter for the seating preference. When you run the load test, the Vusers will use the parameter values instead of the recorded value, **Aisle**.

When you run the script, the replay log shows the parameter substitution that occurs for each iteration. Note that the Vuser chose **Aisle** for the first iteration and **Window** for the second iteration.

How do I verify Web page content?

When running a test, you often need to verify if certain content is found on the returned page. A **content check** verifies that expected information appears on a Web page while the script is running. You can insert two types of content checks:

- **Text check.** Checks that a text string appears on a Web page.
- **Image check.** Checks for an image on a Web page.

Looking for Text

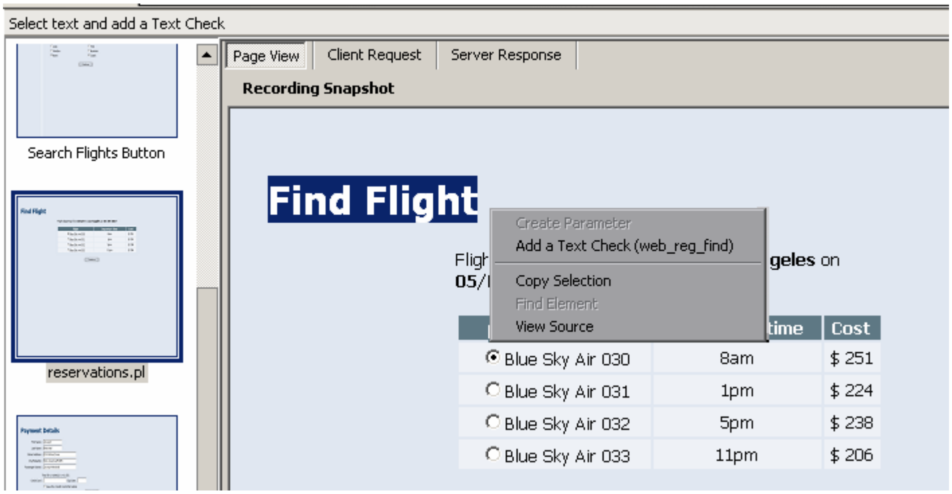
In this section, you will add a text check that checks if the phrase **Find Flight** appears on the reservations page in your script.

To insert a text check:

1 **Open the Content Check wizard.**

Ensure that the Task Pane is displayed. If not, click the **Task** button. In the Task Pane under **Enhancements,** click **Content Checks.**

The Content Check wizard opens displaying thumbnails of each step in the script.



		time	Cost
●	Blue Sky Air 030	8am	$ 251
○	Blue Sky Air 031	1pm	$ 224
○	Blue Sky Air 032	5pm	$ 238
○	Blue Sky Air 033	11pm	$ 206

Select the **Page View** tab in the right pane to display snapshots of the thumbnails.

2 Select the page containing the text you want to check.

Click the fourth thumbnail entitled **reservations.pl**.

3 Select the text that you want to check.

Highlight the words Find Flight within the snapshot. With the words selected, right click and select **Add a Text Check (web-reg-find)**.

The Find Text dialog box opens displaying the text you selected in the **Search for specific Text** box. Click **OK**.

4 View the new step.

In tree view (**View** > **Tree View**) you will notice that VuGen inserts a new step, **Service:Reg Find,** into the script. This step registers the text check—LoadRunner checks for the text after running the step. During replay, VuGen will look for the text **Find Flight** and indicate in the replay log whether or not it was found.

How can I produce debugging information?

At certain points during the test run, you often need to send messages to the output, indicating your location and other information. These output messages will appear in both the replay log and in the Controller's Output window. You can send a standard output message or a message that indicates that an error occurred.

The recommended way to work with error messages is to check for a **Failed** status. If the status is **Failed**, you instruct VuGen to issue an error message. Refer to the examples in the *HP LoadRunner Online Function Reference*.

In this section we will instruct VuGen to insert an output message after the application completes a full booking.

To insert Output messages:

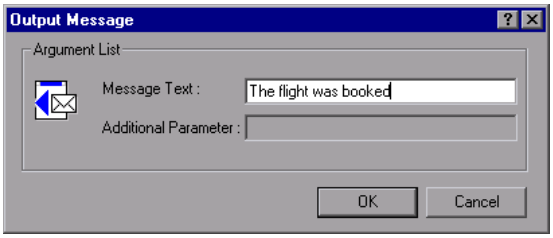
1 Select a location.

Select the last step, **Image: SignOff Button**. The snapshot opens on the right.

2 Insert an Output message.

Select **Insert** > **New Step**. The Add Step dialog box opens. Scroll down, select **Output Message**, and click **OK**.

The Output Message dialog box opens.



Message Text : The flight was booked

Additional Parameter :

3 Type the message.

In the **Message Text** box, type The flight was booked and click **OK**. The Output Message is added to the tree.

4 Save the script.

Note: To insert an error message you would repeat the same process, except that you would select **Error Message** instead of **Output Message** in the Add Step dialog box.

Did my test succeed?

In this section, you will run the enhanced script and view the replay log for text and image checks. You will view the text and image checks, the transactions, and the parameterization.

By default, image checks are disabled during playback since they require more memory. If you want to perform an image check, you need to enable checks in the run-time settings.

1 Enable image checks.

Open the Run-Time settings (**Vuser** > **Run-Time Settings**) and select the **Internet Protocol: Preferences** node. Select **Enable image and text check**. Click **OK** to close the Run-Time Settings dialog box.

2 Run the script.

Click **Run** or select **Vuser** > **Run**. VuGen begins running the script, creating a replay log in the Output window. Wait for the script to finish running.

3 Locate the text check.

Ensure that the Output window is open (**View** > **Output Window**). Click in the **Replay Log** and press CTRL+F to open the Find dialog box. Search for web_reg_find. The first instance says as follows:

```
Registering web_reg_find was successful.
```

This is not the actual text check—it only prepares VuGen to check for the text after the form submission.

Search again (F3) for the next instance of web_reg_find. This instance indicates:

```
Registered web_reg_find successful for “Text=Find Flight” (count=1).
```

This verifies that the text was found. If someone changes the Web page and removes the phrase Find Flight, then in subsequent runs, the output will indicate that the text was not found.

4 Locate the beginning of a transaction.

Click in the replay log and press CTRL+F to open the Find dialog box. Search for the word Transaction. This notification is shown in blue.

5 View the parameter substitution.

Click in the replay log and press CTRL+F to open the Find dialog box. Search for the word Parameter. The log contains a notification “seat” = “Aisle”. Search again (F3) for the next substitution. Note how VuGen takes a different value for each iteration.

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

Where To Go From Here

Now that you have created a script and adapted it for a load test, you can proceed to Lesson 7, “Creating a Load Testing Scenario.”

7

Creating a Load Testing Scenario

In the previous lesson you successfully validated your test in the Virtual User Generator. In this lesson, you will test your application under load. You will emulate the actions of ten travel agents concurrently using the flight reservation system, and observe the behavior of the system under load. To design and run this test, you use the LoadRunner Controller.

In this lesson you will cover the following topics:

- Introducing the LoadRunner Controller
- How Do I Start the Controller?
- The Controller Window at a Glance
- How Do I Modify the Script Details?
- How do I generate a heavy load?
- How do I emulate real load behavior?
- How do I emulate different types of users?
- How do I monitor the system under load?

Introducing the LoadRunner Controller

Load testing means testing your application under typical working conditions, for example, many travel agents reserving flights on the same flight reservation system at the same time.

You design the test to emulate real-life situations. To do this, you need to be able to generate heavy load on an application and schedule when the load is applied (because users do not log on and off the system at precisely the same time). You also need to emulate different kinds of user activity and behavior. For example, some users might be using Netscape instead of Internet Explorer to see the application's performance, and using different network connections such as modem, DSL, or cable. You create and save these settings in a scenario.

The Controller provides you with all the tools you need to help you build and run tests to accurately emulate your working environment.

Scenario Objectives

In this lesson, the objective is to create a scenario that emulates the behavior of ten travel agents simultaneously logging on, searching flights, purchasing flights, checking itineraries, and logging off the system.

How Do I Start the Controller?

To begin creating a scenario, you open the Controller and create a new scenario.

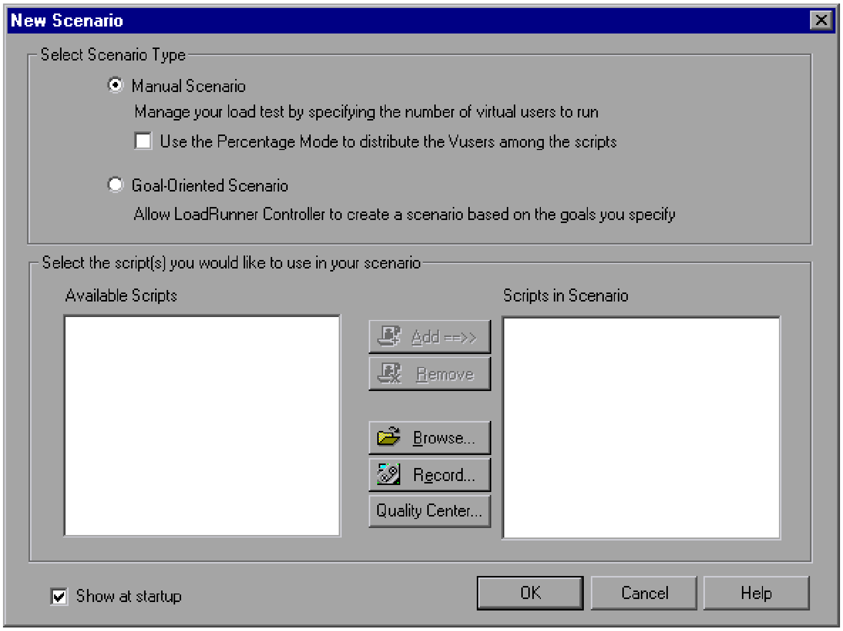
1 Open HP LoadRunner.

Select **Start** > **Programs** > **LoadRunner** > **LoadRunner**. The HP LoadRunner Launcher window opens.

2 Open the Controller.

In the **Load Testing** tab, click **Run Load Tests**. The LoadRunner Controller opens.

By default, the Controller opens with the New Scenario dialog box.



New Scenario

Select Scenario Type

● Manual Scenario

Manage your load test by specifying the number of virtual users to run

□ Use the Percentage Mode to distribute the Vusers among the scripts

○ Goal-Oriented Scenario

Allow LoadRunner Controller to create a scenario based on the goals you specify

Select the script(s) you would like to use in your scenario

☑ Show at startup

3 Select a scenario type.

There are two scenario types:

➤ A **Manual Scenario** gives you control over the number of running Vusers and the times at which they run, and lets you test how many Vusers your application can run simultaneously. You can use the **Percentage Mode** to distribute the total number of Vusers among scripts based on a percentage specified by your business analyst. The **Percentage Mode** check box is selected by default the first time you start LoadRunner following installation. If it is selected now, clear it.

➤ A **Goal-Oriented Scenario** is used to determine if your system can achieve a particular goal. You determine the goal based on, for example, a specified transaction response time or number of hits/transactions per second, and LoadRunner automatically builds a scenario for you based on these goals. You will create a goal-oriented scenario in Lesson 9, "Advanced Goal-Oriented Scenario."

Select **Manual Scenario**.

4 Add a script to the load test.

In this tutorial, you will use just one Vuser script to model a single group of users performing identical actions. To emulate a real scenario with a more varied mix of user profiles, you would create different groups running several scripts with different user settings.

The script that you previously recorded in VuGen contains the business processes that you want to test. They include logging on, searching for a flight, buying a ticket, checking the flight itinerary, and then logging off the site. You will add a similar script to the scenario, and configure the scenario to emulate eight travel agents simultaneously performing these actions on the flight reservation system. You will add two more users during the test itself.

For this purpose, a script is provided that is similar to the one you created. We recommend that you use the sample script.

Click the **Browse** button, and navigate to **basic_script** in the *<LoadRunner Installation>***Tutorial** directory.

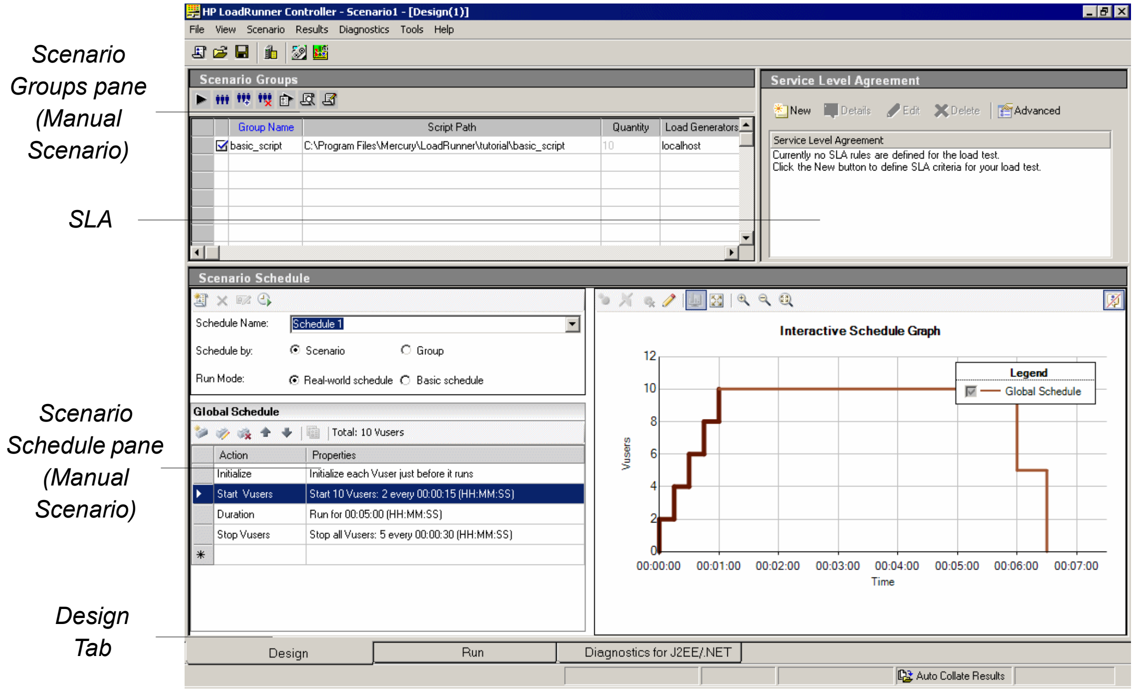
The script is displayed in the Available Scripts section and in the Scripts in Scenario section.

Click **OK**. The LoadRunner Controller opens your scenario in the Design tab.

The Controller Window at a Glance

The Controller window's Design tab is divided into three primary sections:

- Scenario Groups pane
- Service Level Agreement pane
- Scenario Schedule pane



	Group Name	Script Path	Quantity	Load Generators
☑	basic_script	C:\Program Files\Mercury\LoadRunner\tutorial\basic_script	10	localhost

Service Level Agreement

Currently no SLA rules are defined for the load test.
Click the New button to define SLA criteria for your load test.

Schedule Name: Schedule 1

Schedule by: ● Scenario ○ Group

Run Mode: ● Real-world schedule ○ Basic schedule

Global Schedule

Total: 10 Vusers

Action	Properties
Initialize	Initialize each Vuser just before it runs
Start Vusers	Start 10 Vusers: 2 every 00:00:15 (HH:MM:SS)
Duration	Run for 00:05:00 (HH:MM:SS)
Stop Vusers	Stop all Vusers: 5 every 00:00:30 (HH:MM:SS)

- **Scenario Groups pane.** You configure the Vuser groups in the Scenario Groups pane. You create different groups to represent typical users of your system and specify the number of Vusers that will run, and the machine that they will run on.

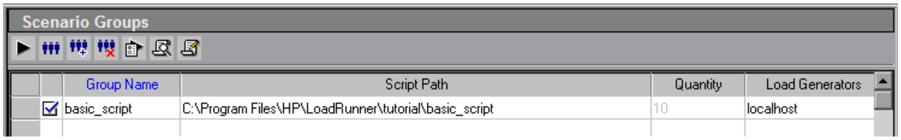
- **Service Level Agreement pane.** When you design a load test scenario, you can define goals or **Service Level Agreements** (SLAs) for the performance metrics. When you run the scenario, LoadRunner gathers and stores performance-related data. When you analyze the run, Analysis compares this data against the SLAs and determines SLA statuses for the defined measurements. You will learn more about SLAs in lesson 10.

- **Scenario Schedule pane.** In the Scenario Schedule pane, you set the load behavior to accurately portray real-world user behavior. You define actions according to which the Vusers will run, the rates at which load is applied to the application, the load test duration, and how the load is stopped.

How Do I Modify the Script Details?

You modify the script details as follows:

1 Check to see that the basic_script appears in the Group Name column of the Scenario Groups pane.

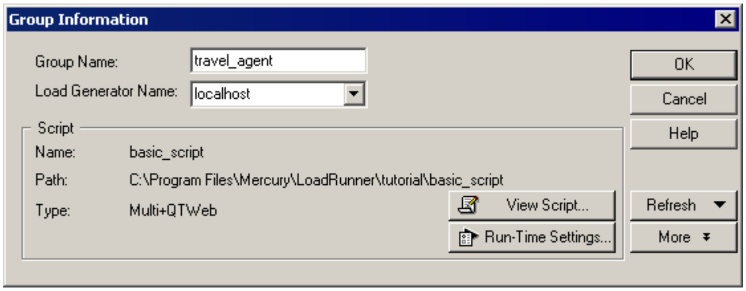


	Group Name	Script Path	Quantity	Load Generators
☑	basic_script	C:\Program Files\HP\LoadRunner\tutorial\basic_script	10	localhost

2 Change the Group Name.

Select the script and click the **Details** button. The Group Information dialog box opens.

In the **Group Name** box, enter a more meaningful name, for example travel_agent.



Group Name: travel_agent

Load Generator Name: localhost

Script

Name: basic_script

Path: C:\Program Files\Mercury\LoadRunner\tutorial\basic_script

Type: Multi+QTWeb

Click **OK**. The name is displayed in the Scenario Groups pane of the Design tab.

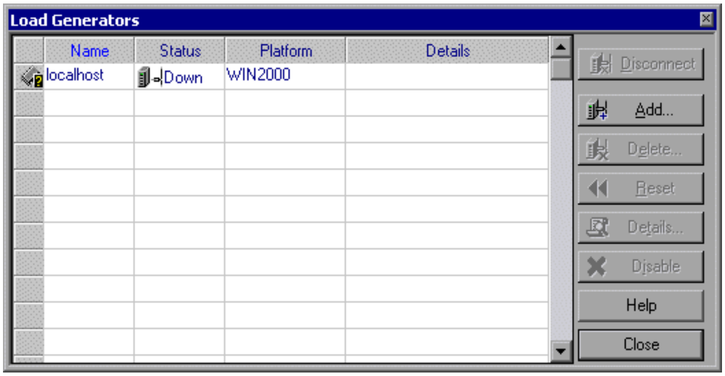
How do I generate a heavy load?

After you have added your scripts, you configure the machines that generate load.

Load generators are the machines that create load on the application by running Vusers. You can use a number of load generators, each hosting many Vusers. In this section, you will learn about adding load generators to the scenario and testing the load generator connection.

1 Add a load generator.

In the Design view, click the **Load Generators** button. The Load Generators dialog box opens, showing details for the load generator called **localhost**.



Load Generators

Name	Status	Platform	Details
localhost	Down	WIN2000	

In this tutorial, you will use your local computer as the load generator (included in the scenario by default). The status of the **localhost** load generator is **Down**. This indicates that the Controller is not connected to the load generator.

Note: In a typical production system, you would have a few load generators, each hosting many Vusers.

---

2 Test the load generator connection.

When you run a scenario, the Controller connects to the load generators automatically. However, you can test the connections before trying to run the scenario.

Select **localhost** and click **Connect**.

The Controller attempts to connect to the load generator machine. When a connection has been made, the status changes from **Down** to **Ready**.

Click **Close**.

How do I emulate real load behavior?

After you have added your load generators, you are ready to configure load behavior.

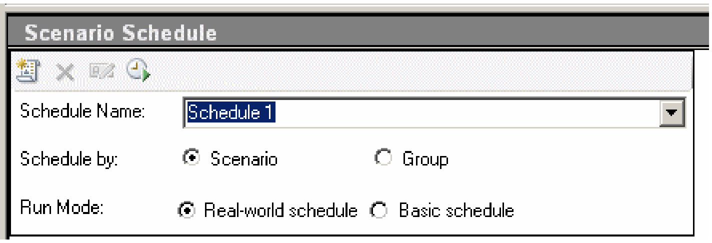
Typical users do not log on and off the system at precisely the same time. LoadRunner allows users to gradually log on to and off the system. It also lets you determine the duration of the scenario, and the way in which it is stopped. The scenario that you will configure below will be relatively simple. However, when designing a scenario that more accurately reflects a real life scenario, you can define more true to life Vuser activity.

You configure the load behavior for a manual scenario in the Scenario Schedule pane of the Controller window. The Scenario Schedule pane is divided into three sections: the Schedule Definition area, the Actions grid, and the Interactive Schedule graph.

You will now change the default load settings and configure a scenario schedule.

1 Select schedule type and run mode.

In the Schedule Definition pane, ensure that **Schedule by: Scenario** and **Run Mode: Real-world schedule** are selected.



Scenario Schedule

Schedule Name: Schedule 1

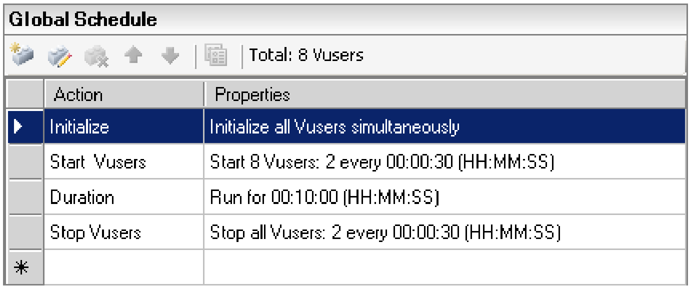
Schedule by: ● Scenario ○ Group

Run Mode: ● Real-world schedule ○ Basic schedule

2 Set the schedule actions definitions.

You can set the Start Vusers, Duration, and Stop Vusers actions for the scenario schedules from either the Actions grid, or by manipulating the Interactive Schedule graph. When you set the definitions from the graph, the Actions grid properties adjust accordingly.

You will now set the definitions so that the Actions grid will look as below.



Global Schedule

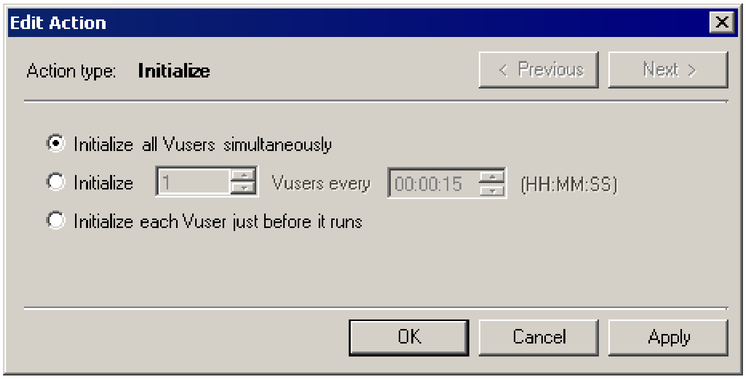
Total: 8 Vusers

Action	Properties
Initialize	Initialize all Vusers simultaneously
Start Vusers	Start 8 Vusers: 2 every 00:00:30 (HH:MM:SS)
Duration	Run for 00:10:00 (HH:MM:SS)
Stop Vusers	Stop all Vusers: 2 every 00:00:30 (HH:MM:SS)

a Set up the Vuser Initialization.

Initialization means preparing the Vusers and the load generators for a load test run by running the **vuser_init** action in the script. Initializing Vusers before they start running reduces CPU consumption and helps provide more realistic results.

Double-click **Initialize** in the Action grid. The Edit Action dialog box opens displaying the Initialize action. Select **Initialize all Vusers simultaneously**.


Edit Action

Action type: **Initialize**

● Initialize all Vusers simultaneously

○ Initialize 1 Vusers every 00:00:15 (HH:MM:SS)

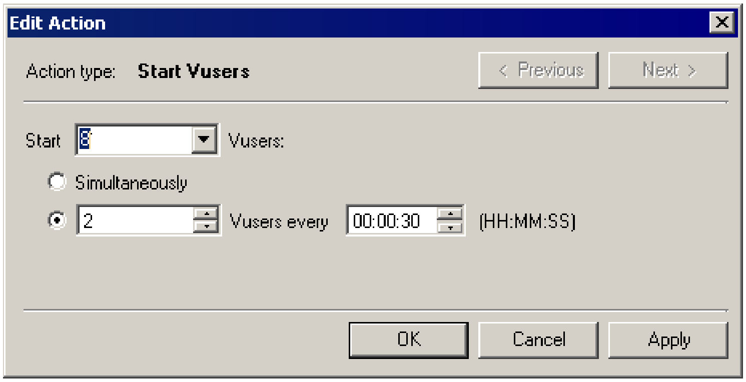
○ Initialize each Vuser just before it runs

b Specify a gradual start (from the Schedule Actions grid).

Starting Vusers at intervals allows you to examine the gradual increase of Vuser load on the application under test over time, and helps you pinpoint exactly when the system response time slows down.

Double-click **Start Vusers** in the Action grid. The Edit Action dialog box opens displaying the Start Vusers action.

In the **Start X Vusers** box, enter 8 Vusers and select the second option—Start 2 Vusers every 30 seconds.


Edit Action

Action type: **Start Vusers**

Start 8 Vusers:

○ Simultaneously

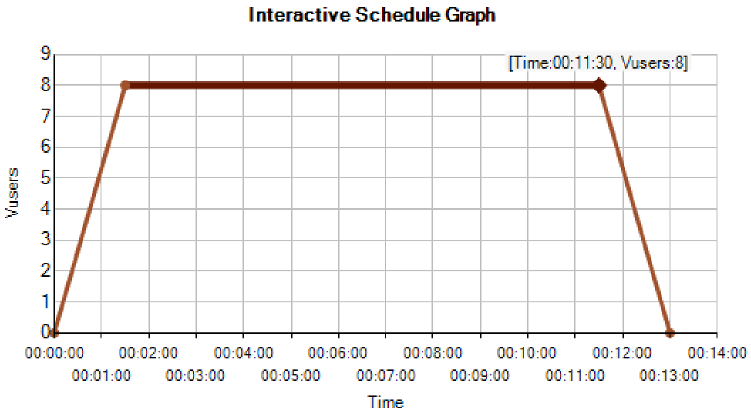
● 2 Vusers every 00:00:30 (HH:MM:SS)

c Schedule the duration (from the Interactive Schedule Graph).

You specify a duration to make sure that the Vusers continue performing the schedule action for a specific period so you can measure continuous load on the server. If you set a duration, the script will run for as many iterations as necessary during that period, disregarding the number of iterations set in the script's run-time settings.

Make sure that the Interactive Schedule Graph is in Edit mode by clicking the Edit Mode button in the Interactive Schedule Graph toolbar.

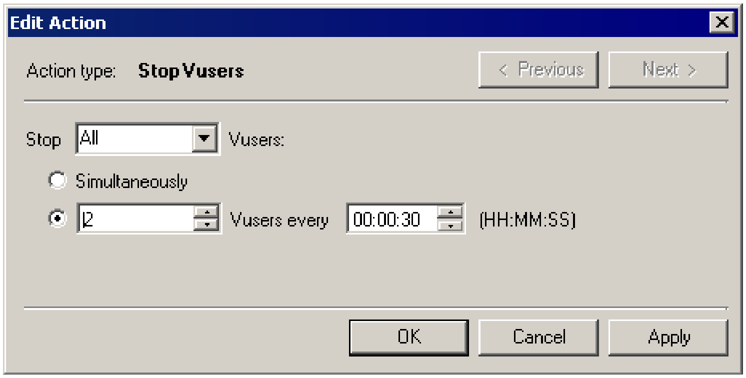
In the Actions grid, click **Duration** or click the horizontal line in the graph which represents Duration. The line is highlighted and a dot and a diamond are displayed at its endpoints. Drag the diamond shaped endpoint to the right until the time in brackets reads **00:11:30**. You have just set the Vusers to run for a duration of 10 minutes.



d Schedule a gradual closure (from the Schedule Actions grid).

Gradually stopping Vusers is recommended to help detect memory leaks and check system recovery, after the application has reached a threshold.

Double-click **Stop Vusers** in the Actions grid. The Edit Action dialog box opens displaying the Stop Vusers action. Select the second option—Stop 2 Vusers every 30 seconds



Edit Action

Action type: **Stop Vusers**

Stop All Vusers:

○ Simultaneously

● 2 Vusers every 00:00:30 (HH:MM:SS)

How do I emulate different types of users?

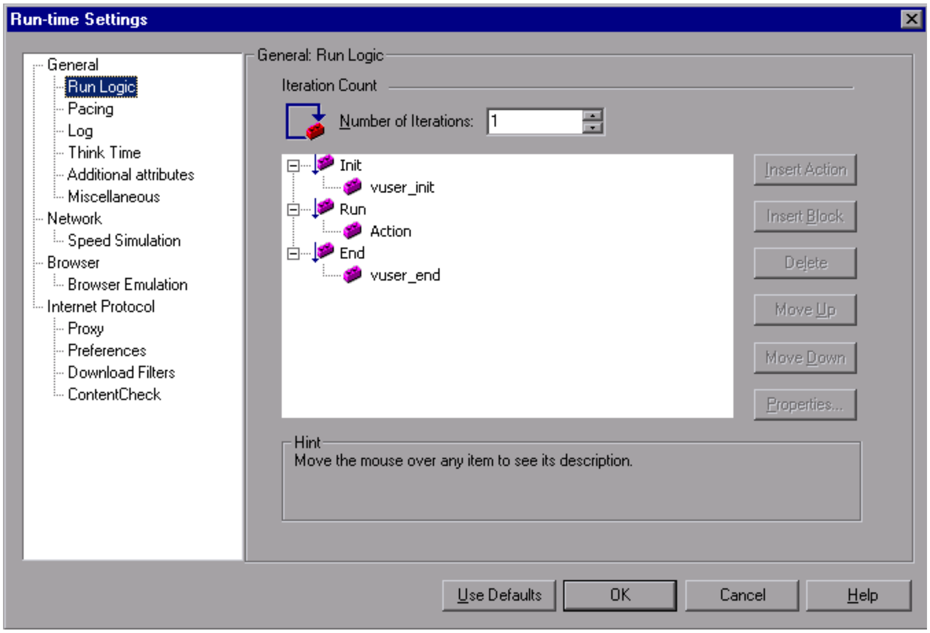
Now that you have configured a load schedule, you will specify how your Vusers will behave during the test.

When emulating a real user, you need to consider the user's actual behavior. Behavior refers to the time that a user takes to pause between actions, the number of times he repeats an action, and so on.

In this section, you will learn more about LoadRunner's run-time settings, and you will enable think time and logging.

1 Open the run-time settings.

In the Design tab, select the script and click the **Run-Time Settings** button. The run-time settings are displayed.



Number of Iterations: 1

Move the mouse over any item to see its description.

The run-time settings let you emulate different kinds of user activity and behavior. They include:

- **Run Logic.** The number of times a user repeats a set of actions.
- **Pacing.** The time to wait before repeating the action.

- **Log.** The level of information that you want to gather during the test.

  The first time you run a scenario, it is recommended to generate log messages to make sure that you have debugging information in case the first run fails.

- **Think Time.** The time the user stops to think between steps.

  Since users interact with the application according to their experience level and objectives, more technically proficient users may work more quickly than new users. Vusers can be made to emulate their real-world counterparts more accurately during a load test by enabling think time.

- **Speed Simulation.** Users using different network connections such as modem, DSL, and cable.

- **Browser Emulation.** Users using different browsers to see their application's performance.

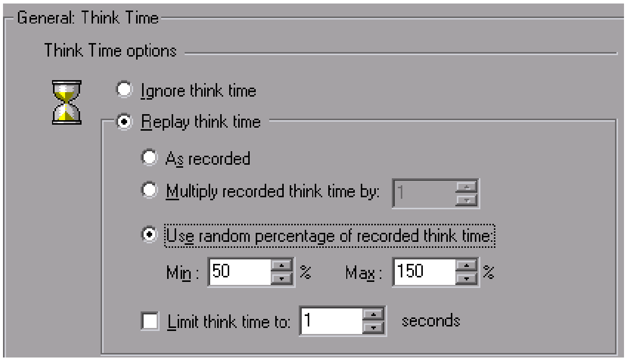
- **Content Check.** For automatically detecting user-defined errors.

  Suppose that your application sends a custom page when an error occurs. This custom page always contains the words **ASP Error**. You need to search all of the pages returned by the server and see if the text **ASP Error** appears.

  You can set up LoadRunner to automatically look for this text during the test run, using the Content Check run-time settings. LoadRunner searches for the text and generates an error if it is detected. During the scenario run, you can clearly identify the content check errors.

2 Enable think time.

Select the **General:Think Time** node. Select **Replay think time**, and select the **Use random percentage of recorded think time** option. Specify a minimum of 50% and a maximum of 150%.



General: Think Time

Think Time options

○ Ignore think time

● Replay think time

○ As recorded

○ Multiply recorded think time by: 1

● Use random percentage of recorded think time:

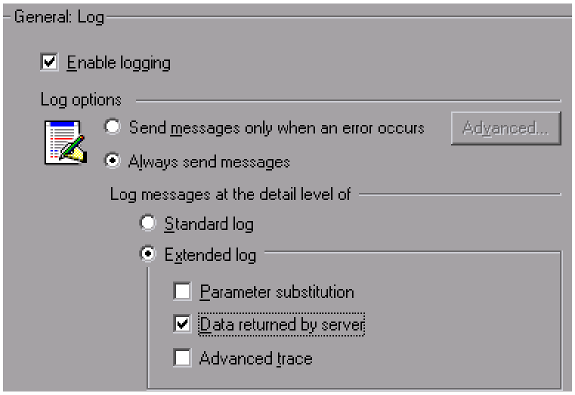
Min : 50 % Max : 150 %

□ Limit think time to: 1 seconds

After you use a random percentage of the recorded think time to emulate users with a varying range of proficiency. For example, if the recorded think time for selecting a flight was 4 seconds, the random think time could be anything between 2-6 seconds (50% to 150% of 4).

3 Enable logging.

Select the **General:Log** node, and select **Enable logging**. In the Log options, select **Always send messages**. Select **Extended log**, and select **Data returned by server**.



General: Log

☑ Enable logging

Log options

○ Send messages only when an error occurs

● Always send messages

Log messages at the detail level of

○ Standard log

● Extended log

□ Parameter substitution

☑ Data returned by server

□ Advanced trace

Note: After the initial debugging run, extended logging is not recommended for a load test. It is only enabled for the purposes of this tutorial to provide information for the Vuser Output log.

Click **OK** to close the Run-Time settings.

How do I monitor the system under load?

Now that you have defined how your Vusers will behave during the test, you are ready to set up your monitors.

While creating heavy load on an application, you want to see how the application performs in real time and where potential bottlenecks exist. You use LoadRunner's suite of integrated monitors to measure the performance of every single tier, server, and component of the system during the load test. LoadRunner includes monitors for a variety of major backend system components including Web, application, database, and ERP/CRM servers.

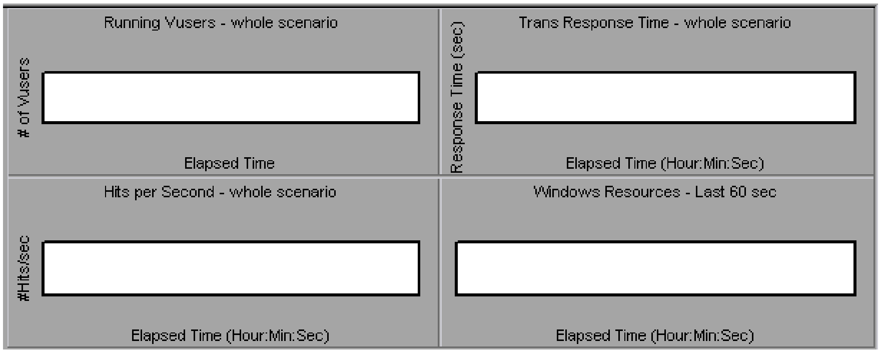
For instance, you can select a Web Server Resources monitor according to the type of Web server that is running. You can purchase a license for the relevant monitor, for example IIS, and use this monitor to pinpoint problems reflected in the IIS resources.

In this section, you will learn how to add and configure the Windows Resources monitor. You can use this monitor to determine the impact of load on your CPU, disk, and memory resources.

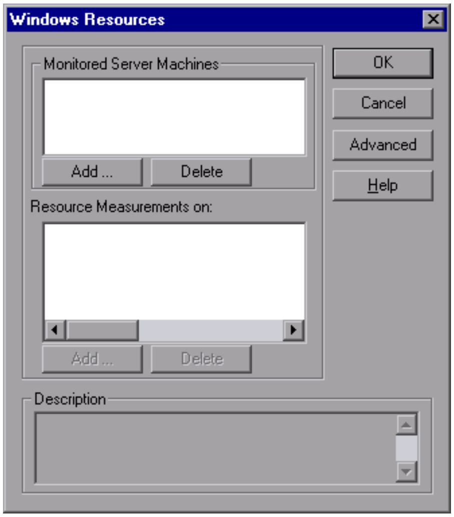
1 Select the Windows Resources Monitor.

Click the **Run** tab in the Controller window to open the Run view.

The Windows Resources graph is one of four default graphs that is displayed in the graph viewing area. You will learn how to open other graphs in the next lesson.

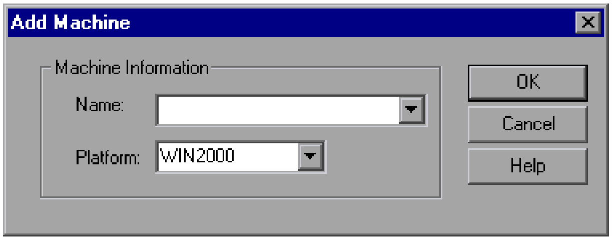

Running Vusers - whole scenario
# of Vusers
Elapsed Time
Trans Response Time - whole scenario
Response Time (sec)
Elapsed Time (Hour:Min:Sec)
Hits per Second - whole scenario
#Hits/sec
Elapsed Time (Hour:Min:Sec)
Windows Resources - Last 60 sec
Elapsed Time (Hour:Min:Sec)

Right-click the Windows Resources graph and select **Add Measurements**. The Windows Resources dialog box opens.



2 Select the monitored server.

In the **Monitored Server Machines** section of the Windows Resources dialog box, click **Add**. The Add Machine dialog box opens.


Add Machine

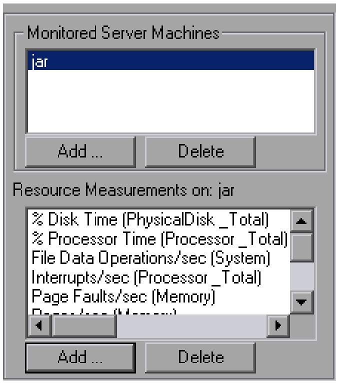
Machine Information

Name:

Platform: WIN2000

In the **Name** box type localhost. (If your load generator was running off a different machine you would type the server name or IP address of that machine.) In the **Platform** box, enter the platform on which the machine runs. Click **OK**.

The default Windows Resources measurements are listed under the **Resource Measurements on <server machine>**.


Monitored Server Machines

jar

Resource Measurements on: jar

% Disk Time (PhysicalDisk _Total)
% Processor Time (Processor _Total)
File Data Operations/sec (System)
Interrupts/sec (Processor _Total)
Page Faults/sec (Memory)

3 Activate the monitor.

Click **OK** in the Windows Resources dialog box to activate the monitor.

Where To Go From Here

Now that you have designed a load test scenario, you can proceed to Lesson 8, "Running The Load Test."

8

Running The Load Test

When you run the test, LoadRunner creates load on the application. You can then use LoadRunner's monitors and graphs to observe the performance of the application under real-life conditions.

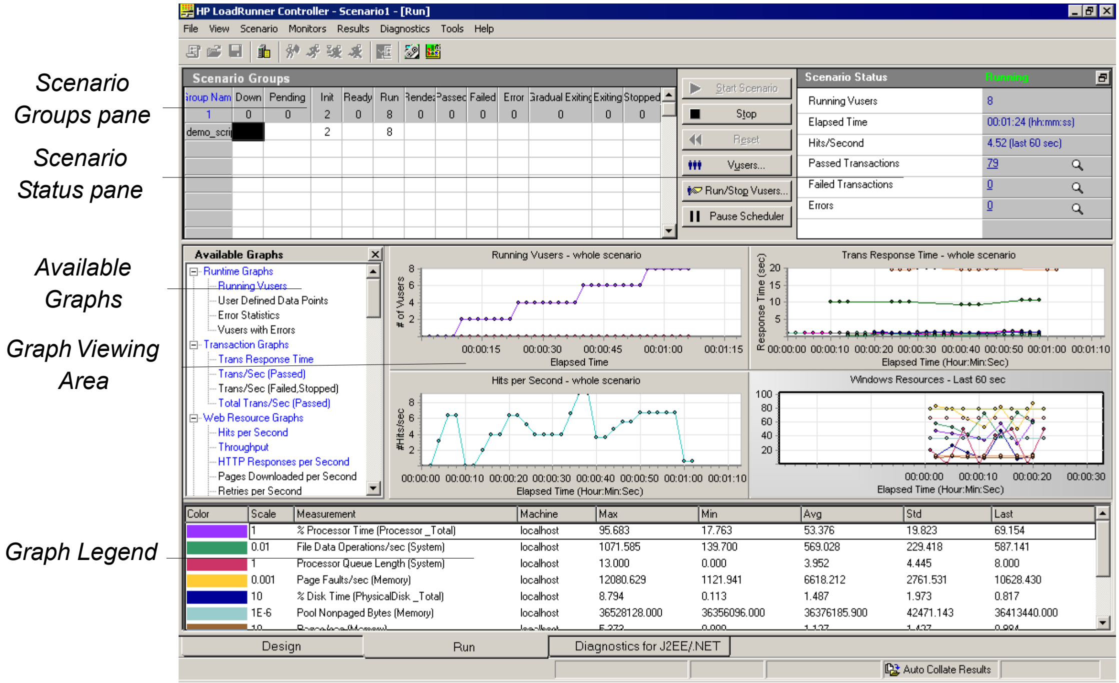
In this lesson you will cover the following topics:

- The Controller Run View at a Glance
- How do I run a load test scenario?
- How do I monitor the application under load?
- How do I watch a Vuser running in real time?
- Where can I view a summary of user actions?
- How can I increase the load during the test?
- How is the application coping under load?
- Did the application encounter errors?
- How do I know that the test has finished running?
- Did the application perform well under load?

The Controller Run View at a Glance

The Run tab in the Controller window is the control center from which the test is managed and monitored. The Run view contains five main sections:

- Scenario Groups pane
- Scenario Status pane
- Available Graphs Tree
- Graph Viewing Area
- Graph Legend



Scenario Groups

Group Nam	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
1	0	0	2	0	8	0	0	0	0	0	0	0
demo_scri			2		8							

Scenario Status Running

Running Vusers	8
Elapsed Time	00:01:24 (hh:mm:ss)
Hits/Second	4.52 (last 60 sec)
Passed Transactions	79
Failed Transactions	0
Errors	0

Color	Scale	Measurement	Machine	Max	Min	Avg	Std	Last
🟪	1	% Processor Time (Processor _Total)	localhost	95.683	17.763	53.376	19.823	69.154
🟩	0.01	File Data Operations/sec (System)	localhost	1071.585	139.700	569.028	229.418	587.141
🟥	1	Processor Queue Length (System)	localhost	13.000	0.000	3.952	4.445	8.000
🟨	0.001	Page Faults/sec (Memory)	localhost	12080.629	1121.941	6618.212	2761.531	10628.430
🟦	10	% Disk Time (PhysicalDisk _Total)	localhost	8.794	0.113	1.487	1.973	0.817
🟦	1E-6	Pool Nonpaged Bytes (Memory)	localhost	36528128.000	36356096.000	36376185.900	42471.143	36413440.000

- **Scenario Groups pane.** In the upper-left pane, you can view the status of Vusers in the scenario groups. You use the buttons to the right of this pane to start, stop, and reset the scenario, to view individual Vuser status, and to increase the load on the application during a scenario by manually adding more Vusers.

- **Scenario Status pane.** In the upper-right pane, you can view a summary of the load test, including the number of running Vusers and the status of each Vuser action.
- **Available Graphs Tree.** In the middle-left pane, you can see a list of the LoadRunner graphs. To open a graph, select a graph in the tree, and drag it into the graph viewing area.
- **Graph Viewing Area.** In the middle-right pane, you can customize the display to view between one and eight graphs (**View** > **View Graphs**).
- **Graph Legend.** In the lower pane, you can view data from the selected graph.

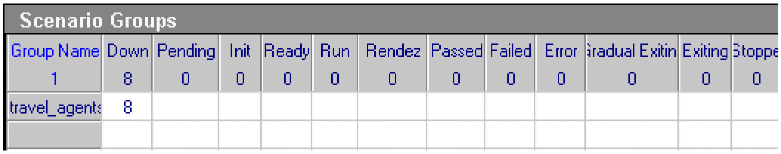
How do I run a load test scenario?

In this section, you will start the scenario.

1 Open the Controller Run view.

Select the **Run** tab at the bottom of the screen.

Notice that there are 8 Vusers in the Down column of the Scenario Groups pane. These are the Vusers you created when you created the scenario.



Scenario Groups

Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	iradual Exitin	Exiting	Stoppe
1	8	0	0	0	0	0	0	0	0	0	0	0
travel_agent:	8											

Since the scenario has not yet run, all other counters remain at zero and all the graphs in the graph viewing area (except Windows Resources) are blank. When you start running the scenario in the next step, the graphs and counters will begin to display information.

2 Start the scenario.

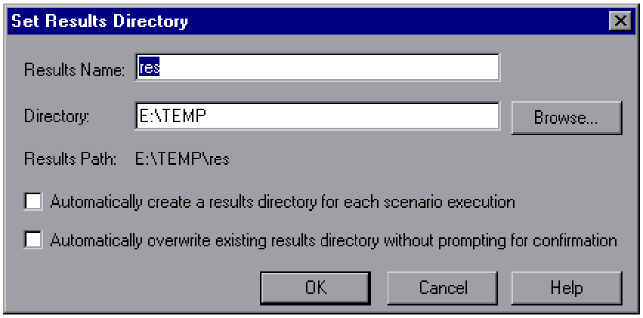
▶

Click the **Start Scenario** button or select **Scenario** > **Start** to begin running the test.

If you are running the tutorial for the first time, the Controller begins the scenario. The result files are saved automatically to the load generator's temp directory.

If you are repeating the test, you will be prompted to overwrite the existing results file. Click **No**, because the results of the first load test should be used as baseline results to be compared with subsequent load test results.

The Set Results Directory dialog box opens.



Set Results Directory

Results Name: res

Directory: E:\TEMP

Results Path: E:\TEMP\res

□ Automatically create a results directory for each scenario execution

□ Automatically overwrite existing results directory without prompting for confirmation

Specify a new results directory. Enter a unique and meaningful name for each results set, because you may want to superimpose the results of several scenario runs when you come to analyze the graphs.

How do I monitor the application under load?

You use the Controller's online graphs to view performance data collected by the monitors. You use this information to isolate potential problem areas in your system environment.

1 Examine the performance graphs.

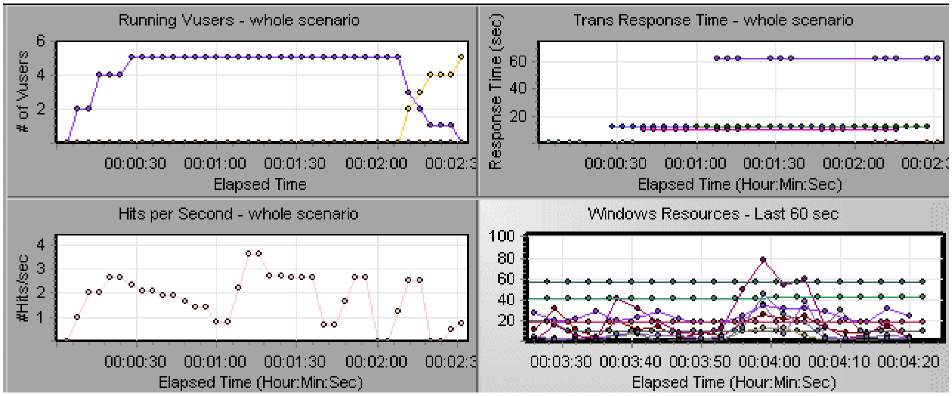
The Run tab displays the following default online graphs:

➤ **Running Vusers - Whole Scenario graph.** Displays the number of Vusers running at a given time.

➤ **Transaction Response Time - Whole Scenario graph.** Shows the amount of time it takes for each transaction to be completed.

➤ **Hits per Second - Whole Scenario graph.** Displays the number of hits (HTTP requests) made to the Web server by Vusers during each second of the scenario run.

➤ **Windows Resources graph.** Displays the Windows resources measured during a scenario.



2 Highlight individual measurements.

Double-click the Windows Resources pane to enlarge it. Notice that each measurement appears on a color-coded row in the graph legend. Each row corresponds to a line in the graph with the same color.

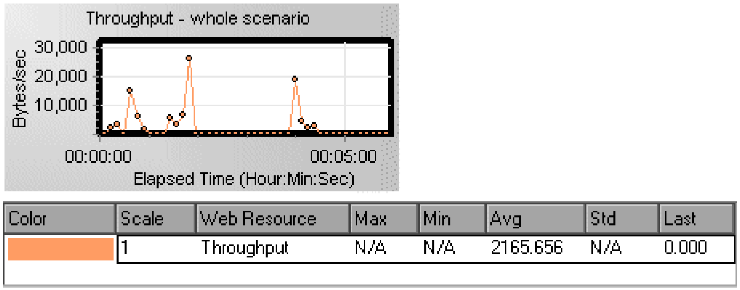
Selecting a row highlights the corresponding line in the graph, and vice versa. Double-click the graph again to reduce it.

3 View the throughput information.

Select the **Throughput** graph in the Available Graphs tree, and drag it into the graph viewing area. The Throughput graph measurements are displayed in the display window and the graph legend.

The Throughput graph shows the amount of data (measured in bytes) that the Vusers receive from the server at any given second. You can compare this graph with the Transaction Response Time graph to see how throughput affects transaction performance.

If the throughput scales upward as time progresses and the number of Vusers increases, this indicates that the bandwidth is sufficient. If the graph were to remain relatively flat as the number of Vusers increased, it would be reasonable to conclude that the bandwidth is constraining the volume of data delivered.



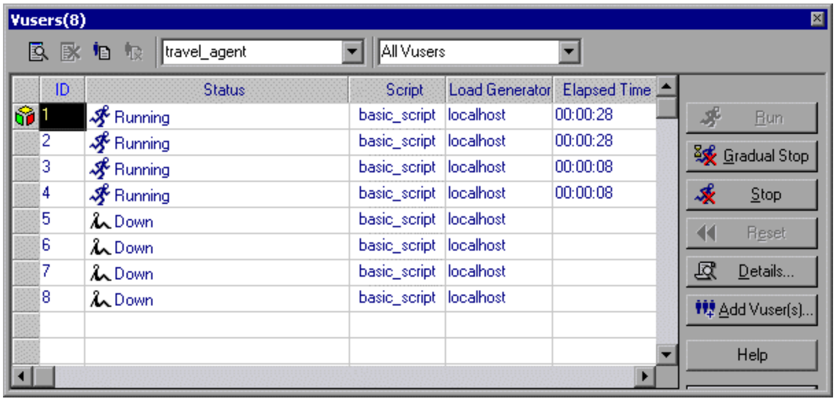
Color	Scale	Web Resource	Max	Min	Avg	Std	Last
🟧	1	Throughput	N/A	N/A	2165.656	N/A	0.000

How do I watch a Vuser running in real time?

When emulating users, you should be able to view their actions in real time and make sure they are performing the right steps. The Controller lets you view the actions in real time using the *run-time viewer.*

To visually observe the Vuser's actions:

1 Click the **Vusers** button. The Vusers window opens.



travel_agent

All Vusers

ID	Status	Script	Load Generator	Elapsed Time
1	Running	basic_script	localhost	00:00:28
2	Running	basic_script	localhost	00:00:28
3	Running	basic_script	localhost	00:00:08
4	Running	basic_script	localhost	00:00:08
5	Down	basic_script	localhost	
6	Down	basic_script	localhost	
7	Down	basic_script	localhost	
8	Down	basic_script	localhost	

The **Status** column displays the status of each Vuser. In the example above, you can see that four Vusers are running and four are in down. The Start Vusers action in the scheduler instructed the Controller to release two Vusers at a time. As the scenario progresses, Vusers will continue to be added in groups of two at 30-second intervals.

2 Select a running Vuser in the Vuser list.

3 Click the **Show the selected Vusers** button on the Vusers toolbar. The Run-Time Viewer opens and displays the action currently being performed by the selected Vuser. The window is updated as the Vuser proceeds through the steps of the recorded script.

4 Click the **Hide the selected Vusers** button on the Vusers toolbar to close the Run-Time Viewer log.

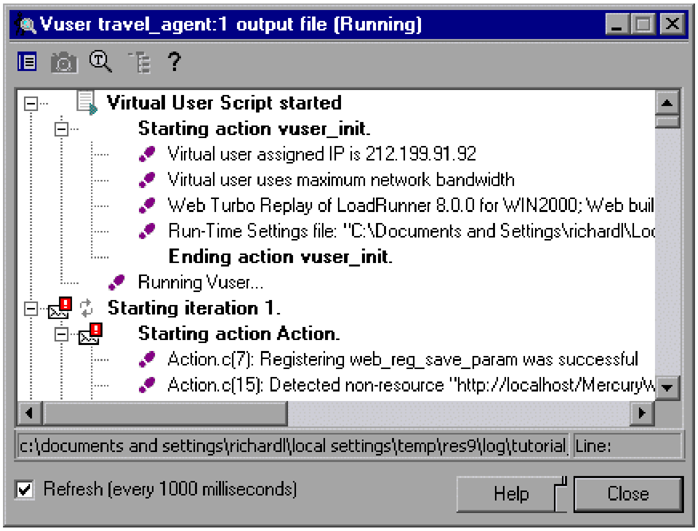
Where can I view a summary of user actions?

To check the progress of an individual Vuser during a running test, you can view a log file containing a text summary of the Vuser's actions.

To review a text summary of the events:

1 Select a running Vuser in the Vusers window, and click the **Show Vuser Log** button.

The Vuser log window opens.



Vuser travel_agent:1 output file (Running)

```
Virtual User Script started
    Starting action vuser_init.
        Virtual user assigned IP is 212.199.91.92
        Virtual user uses maximum network bandwidth
        Web Turbo Replay of LoadRunner 8.0.0 for WIN2000; Web buil
        Run-Time Settings file: "C:\Documents and Settings\richardl\Loc
        Ending action vuser_init.
    Running Vuser...
Starting iteration 1.
    Starting action Action.
        Action.c(7): Registering web_reg_save_param was successful
        Action.c(15): Detected non-resource "http://localhost/MercuryW
```

c:\documents and settings\richardl\local settings\temp\res9\log\tutorial Line:

☑ Refresh (every 1000 milliseconds)

The log contains messages that correspond to the actions of the Vuser. For example, in the window above, the message **Virtual User Script started** indicates the start of the scenario. Scroll to the bottom of the log and watch as new messages are added for each action performed by the selected Vuser.

2 Close the Vuser log window and the Vusers window.

How can I increase the load during the test?

You can increase the load on the application during a running load test by manually adding more Vusers.

To increase load during a load test:

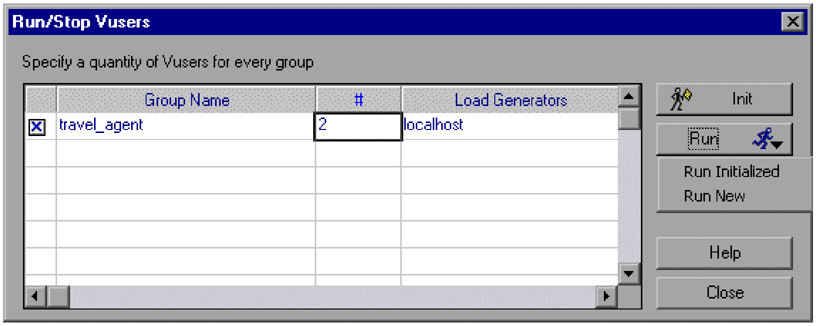
Run/Stop Vuser

1 In the Run view, click the **Run/Stop Vusers** button. The Run/Stop Vuser dialog box opens displaying the number of Vusers currently assigned to run in the scenario.

2 In the # column, enter the number of additional Vusers to the group that you want to add. To run two additional Vusers, replace the number 8 with the number 2, in the # column.

3 Click **Run** to add the Vusers.

If some of the Vusers have not yet been initialized, the **Run Initialized** and **Run New** options open. Select the **Run New** option.



Run/Stop Vusers

Specify a quantity of Vusers for every group

	Group Name	#	Load Generators
☒	travel_agent	2	localhost

These 2 additional Vusers are distributed to the **travel_agent** group and are run on the **localhost** load generator. The Scenario Status pane shows that there are now 10 running Vusers.

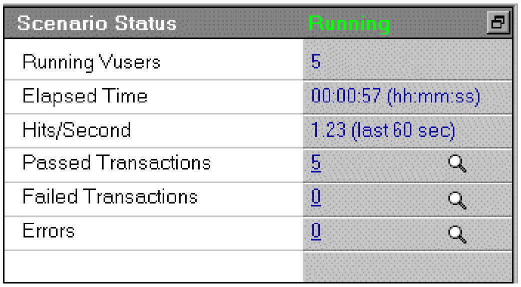
Note: You may get a warning message that LoadRunner Controller cannot activate additional Vusers. This is caused by the fact that you are using your local machine as a load generator and it has very limited memory resources. In most situations you would use a dedicated machine as a load generator to avoid such problems.

How is the application coping under load?

Check the Scenario Status pane for a synopsis of the running scenario, and drill down to see which Vuser actions are causing the application problems. A high proportion of failed transactions and errors indicates that the application is not performing as expected under load.

1 View the test status.

The Scenario Status pane displays the overall status of the scenario.

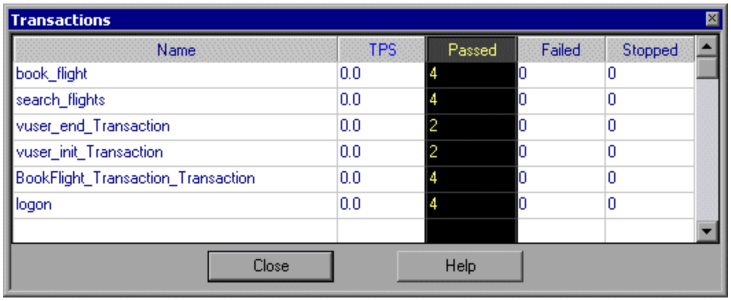


Scenario Status Running

Running Vusers	5
Elapsed Time	00:00:57 (hh:mm:ss)
Hits/Second	1.23 (last 60 sec)
Passed Transactions	5
Failed Transactions	0
Errors	0

2 View breakdown of Vuser actions.

Click **Passed Transactions** in the Scenario Status pane to view a list of transaction details. The Transactions dialog box opens.



Transactions

Name	TPS	Passed	Failed	Stopped
book_flight	0.0	4	0	0
search_flights	0.0	4	0	0
vuser_end_Transaction	0.0	2	0	0
vuser_init_Transaction	0.0	2	0	0
BookFlight_Transaction_Transaction	0.0	4	0	0
logon	0.0	4	0	0

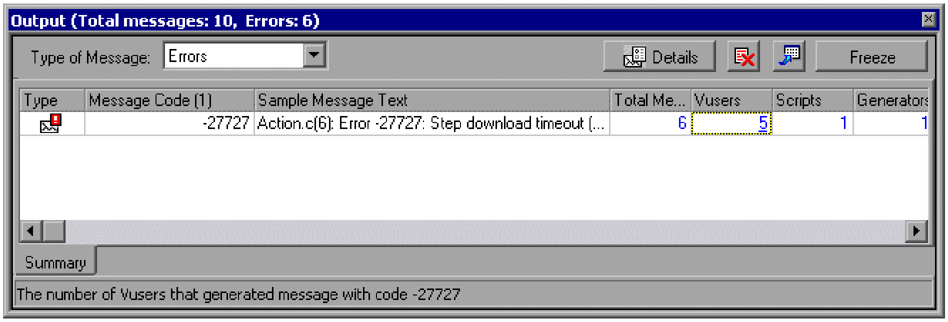
Did the application encounter errors?

If an application starts to fail under heavy load, you are likely to encounter errors and failed transactions. The Controller displays error messages in the Output window.

1 Check for any error messages.

Select **View** > **Show Output** or click **Errors** in the Scenario Status pane.

The Output dialog box opens and lists a message text, the total number of messages generated, the Vusers and load generators that generated the error, and the scripts in which the errors occurred.



Output (Total messages: 10, Errors: 6)

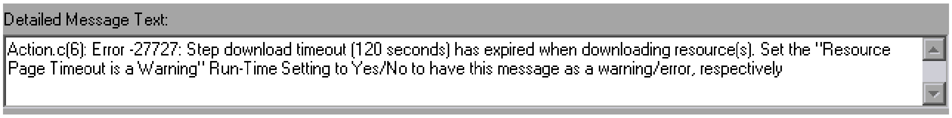
Type of Message: Errors

Type	Message Code (1)	Sample Message Text	Total Me...	Vusers	Scripts	Generators
	-27727	Action.c(6): Error -27727: Step download timeout (...	6	5	1	1

The number of Vusers that generated message with code -27727

To view a message in detail, select the message and click **Details**. The Detailed Message Text box opens, displaying the complete message text.

The example below shows a timeout error. The Web server did not respond to a request within a given time period.



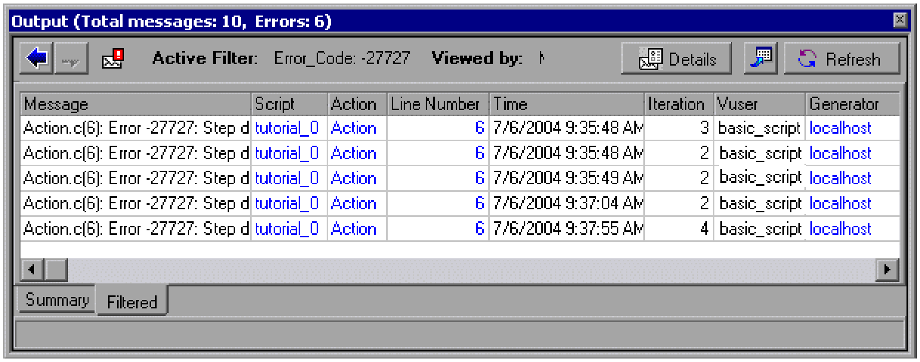
Detailed Message Text:

Action.c(6): Error -27727: Step download timeout (120 seconds) has expired when downloading resource(s). Set the "Resource Page Timeout is a Warning" Run-Time Setting to Yes/No to have this message as a warning/error, respectively

2 View log information details.

You can view information about each message, Vuser, script, and load generator associated with an error code by clicking the blue link in the appropriate column.

For example, to locate where in the script an error occurred, drill down on the **Total Messages** column. The Output window displays a list of all messages of the selected error code, including the time, iteration number, and line in the script where the error occurred.



Output (Total messages: 10, Errors: 6)

Active Filter: Error_Code: -27727 **Viewed by:** M

Message	Script	Action	Line Number	Time	Iteration	Vuser	Generator
Action.c(6): Error -27727: Step d	tutorial_0	Action	6	7/6/2004 9:35:48 AM	3	basic_script	localhost
Action.c(6): Error -27727: Step d	tutorial_0	Action	6	7/6/2004 9:35:48 AM	2	basic_script	localhost
Action.c(6): Error -27727: Step d	tutorial_0	Action	6	7/6/2004 9:35:49 AM	2	basic_script	localhost
Action.c(6): Error -27727: Step d	tutorial_0	Action	6	7/6/2004 9:37:04 AM	2	basic_script	localhost
Action.c(6): Error -27727: Step d	tutorial_0	Action	6	7/6/2004 9:37:55 AM	4	basic_script	localhost

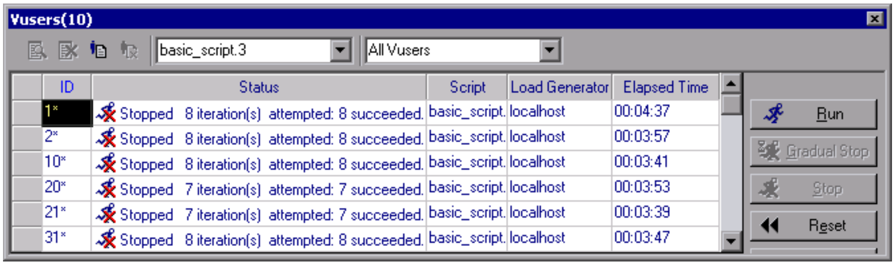
Drill down on the **Line Number** column.

VuGen opens displaying the line in the script at which the error occurred. You can use this information to identify transactions with slow response times that are causing the application to fail under load.

How do I know that the test has finished running?

At the conclusion of the test run, the Scenario Status pane shows the **Down** status. This indicates that the Vusers have stopped running.

You can look in the Vuser dialog box to see the status of each individual Vuser. LoadRunner displays the number of times a Vuser repeated a task (iteration), the number of successful iterations, and the elapsed time.



basic_script.3

All Vusers

ID	Status	Script	Load Generator	Elapsed Time
1*	Stopped 8 iteration(s) attempted: 8 succeeded.	basic_script.	localhost	00:04:37
2*	Stopped 8 iteration(s) attempted: 8 succeeded.	basic_script.	localhost	00:03:57
10*	Stopped 8 iteration(s) attempted: 8 succeeded.	basic_script.	localhost	00:03:41
20*	Stopped 7 iteration(s) attempted: 7 succeeded.	basic_script.	localhost	00:03:53
21*	Stopped 7 iteration(s) attempted: 7 succeeded.	basic_script.	localhost	00:03:39
31*	Stopped 8 iteration(s) attempted: 8 succeeded.	basic_script.	localhost	00:03:47

Did the application perform well under load?

To see how well the application performed under load, you need to look at the transaction response time and determine whether the transaction was within an acceptable limit for the customer. If the transaction response time degrades, you need to look for bottlenecks. You will learn more about this in Lesson 10, "Analyzing Your Scenario."

Once a problem has been isolated, a corroborative effort involving developers, DBAs, network, and other systems experts is required to fix the bottleneck. After adjustments are made, the load test is repeated to confirm that the adjustments had the desired effect. You repeat this cycle to optimize system performance.

To save the scenario so that you can run it again with the same settings, select **File** > **Save** or click the **Save** button, and enter a scenario name in the File name box.

Where To Go From Here

Now that you have learned how to run and view a simple load test scenario, you can proceed to Lesson 9, "Advanced Goal-Oriented Scenario."

9

Advanced Goal-Oriented Scenario

In the previous two lessons you learned how to manually create and run a load test. In this lesson, you will define a goal that you want your test to achieve.

Before deploying an application, you want to run an acceptance test to make sure that the system will stand up to the expected real-life workload. You have a rate at which you expect the server to perform, defined, for example, by the number of hits or transactions per second. This rate might be determined by a business analyst defining the requirements for the application, or it might be taken from previous versions of the application in production, or other sources. You set a goal for the number of hits per second, transactions per second, or the transaction response time that you want to generate, and LoadRunner automatically generates the required goal for you, using the goal-oriented scenario. While the application is under constant load, you can monitor the transaction response time to see the level of service provided to the customer.

In this lesson, you will create a goal-oriented scenario to generate three hits per second on your Web server, and to maintain this level of load for five minutes using a minimum of five Vusers and a maximum of ten. In this lesson you will cover the following topics:

- Which goal type should I use?
- How do I create a goal-oriented scenario?
- The Controller Window at a Glance (Goal-Oriented Scenario)
- How do I define the goal?
- How do I determine load behavior?
- Which online graphs should I monitor?
- How do I run a goal-oriented scenario?
- Did I achieve my goal?

Which goal type should I use?

LoadRunner provides you with five different types of goals in a goal-oriented scenario: the number of concurrent Vusers, the number of hits per second, the number of transactions per second, the number of pages per minute, or the transaction response time that you want your scenario to reach.

- If you know the total number of Vusers that can run your various business processes, you can use a Vusers goal type.
- If you know the strength of your server, you can use a Hits per Second, Pages per Minute, or Transactions per Second goal type.
- If you know the desired response time for completing a transaction, you can use a Transaction Response Time goal type. For example, if you do not want a customer to wait more than five seconds to log in to your e-commerce site, specify a maximum acceptable transaction response time of five seconds, and see how many actual Vusers can be served.

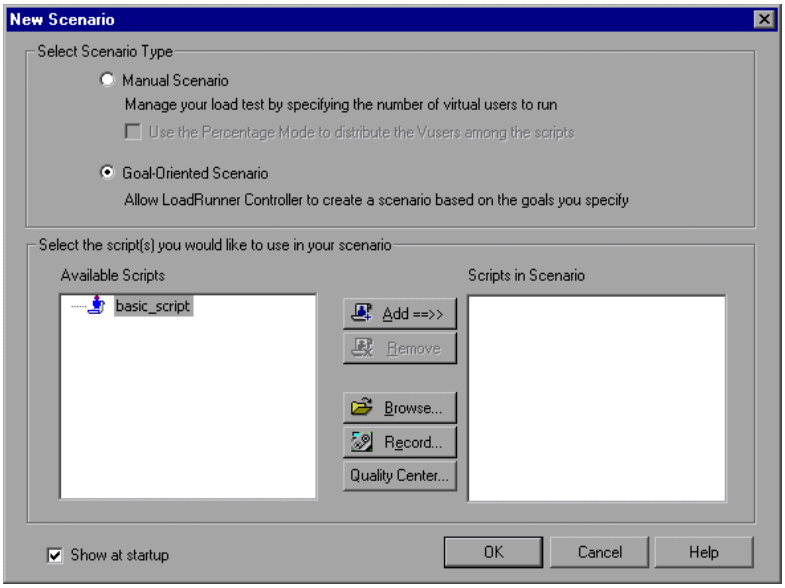
How do I create a goal-oriented scenario?

To emulate a real-life system with a mix of user profiles, you can assign several scripts to the scenario and assign a percentage of the load between them. You should set the percentage according to the expected load.

In this tutorial, you will use just one Vuser script to model a single group of users performing identical actions.

1 Create a new scenario.

Select **File** > **New**, or click **New,** to open the New Scenario dialog box.



New Scenario

Select Scenario Type

○ Manual Scenario

Manage your load test by specifying the number of virtual users to run

☐ Use the Percentage Mode to distribute the Vusers among the scripts

● Goal-Oriented Scenario

Allow LoadRunner Controller to create a scenario based on the goals you specify

Select the script(s) you would like to use in your scenario

Available Scripts: basic_script

Scripts in Scenario

☑ Show at startup

2 Select the scenario type.

Select **Goal-Oriented Scenario**.

3 Select a script.

Select **basic_script** from the Available Scripts list, and click the **Add** button. The script is displayed in the Scripts in Scenario pane.

Click **OK**. The LoadRunner Controller design view opens displaying the **basic_script** in the Script Name column.

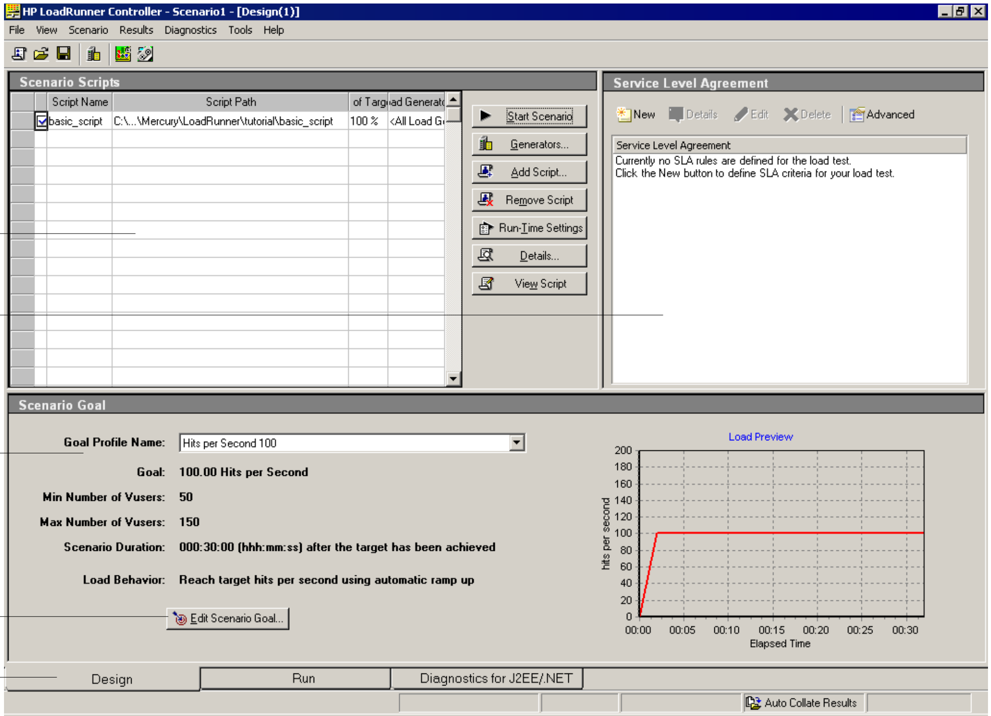
The Controller Window at a Glance (Goal-Oriented Scenario)

The Controller window (goal-oriented) Design view is divided into three primary sections:

➤ Scenario Scripts pane

➤ Service Level Agreement pane

➤ Scenario Goal pane



HP LoadRunner Controller - Scenario1 - [Design(1)]

	Script Name	Script Path	of Targ	ad Generat
☑	basic_script	C:\...\Mercury\LoadRunner\tutorial\basic_script	100 %	<All Load G

Service Level Agreement

Currently no SLA rules are defined for the load test.
Click the New button to define SLA criteria for your load test.

Goal Profile Name: Hits per Second 100

Goal: 100.00 Hits per Second

Min Number of Vusers: 50

Max Number of Vusers: 150

Scenario Duration: 000:30:00 (hhh:mm:ss) after the target has been achieved

Load Behavior: Reach target hits per second using automatic ramp up

Scenario Scripts section

SLA

Scenario Goal section

Edit Scenario Goal

Design Tab

Scenario Scripts pane. In this pane, you determine the Vuser scripts, their paths, the percentage of the total target assigned to each script, and the load generators. You configure the scenario settings from here.

Service Level Agreement pane. When you design a load test scenario, you can define goals or **service level agreements** (SLAs) for the performance metrics. When you run the scenario, LoadRunner gathers and stores performance-related data. When you analyze the run, Analysis compares this data against the SLAs and determines SLA statuses for the defined measurements.

Scenario pane. In the lower pane, you can see your testing goal, the number of users that you want to use to reach that goal, the scenario duration, and load behavior. You define the goal settings from the Edit Scenario Goal dialog box.

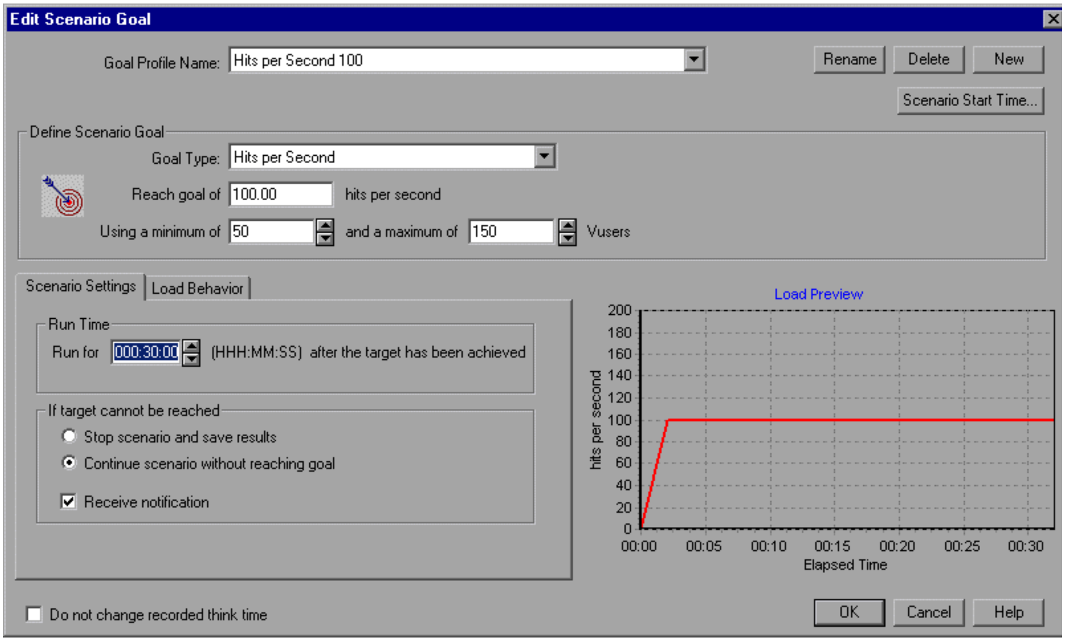
How do I define the goal?

Now that you have selected a script to run, you need to define the goal that you want to achieve.

In this section, you will create a goal profile, and define the scenario goal.

1 Open the Edit Scenario Goal dialog box.

Click the **Edit Scenario Goal** button or select **Scenario** > **Goal Definition**. The Edit Scenario Goal dialog box opens.



Edit Scenario Goal

Goal Profile Name: Hits per Second 100

Define Scenario Goal

Goal Type: Hits per Second

Reach goal of 100.00 hits per second

Using a minimum of 50 and a maximum of 150 Vusers

Scenario Settings | Load Behavior

Run Time

Run for 000:30:00 (HHH:MM:SS) after the target has been achieved

If target cannot be reached

○ Stop scenario and save results

● Continue scenario without reaching goal

☑ Receive notification

□ Do not change recorded think time

2 Specify a logical name for the Goal Profile.

Click **New**, type the new goal profile name (for example: Hits per Second 3) in the New Goal Profile dialog box, and click **OK**.

The new goal profile name appears in the selector.

3 Define the scenario goal.

a In the **Goal Type** box, select **Hits per Second**.

b In the **Reach Goal of X Hits per Second** box, enter **3**.

c Set the minimum-maximum range of Vusers for LoadRunner to run.

Enter **5** as the minimum and **10** as the maximum number of Vusers. This corresponds with the minimum-maximum range of travel agents that you want to create hits on the server simultaneously.

How do I determine load behavior?

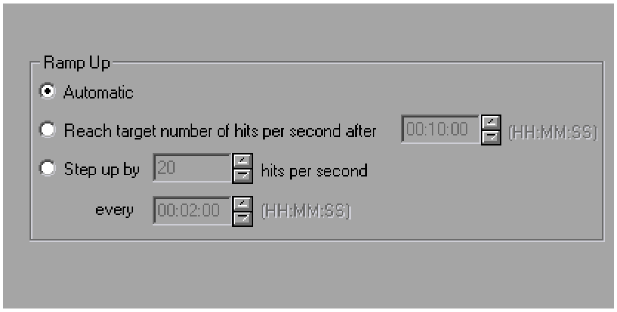
Now that you have defined the test goal, you need to specify how and when you want the Controller to reach your target.

Users do not log on and off the system at precisely the same time. To emulate real users, LoadRunner provides the capability in the Load Behavior tab for users to gradually log on and off the system. You also want the server to remain under the load for a period of time. LoadRunner lets you specify in the Scenario Settings tab the time that the server remains under load.

To define load test behavior:

1 Configure the test to run the Vusers simultaneously.

In the Edit Scenario Goal dialog box, select the **Load Behavior** tab, and select **Automatic**.



Ramp Up

● Automatic

○ Reach target number of hits per second after 00:10:00 (HH:MM:SS)

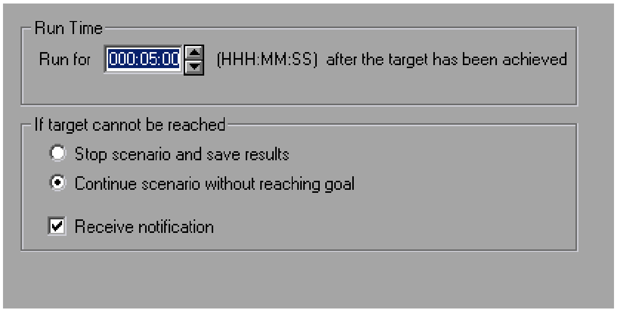
○ Step up by 20 hits per second

every 00:02:00 (HH:MM:SS)

This instructs the Controller to run the required number of Vusers simultaneously.

2 Select the scenario settings.

In the Scenario Settings tab, specify that the test should run for 5 minutes after the target has been achieved, and select **Continue scenario without reaching goal**.



Run Time

Run for 000:05:00 (HHH:MM:SS) after the target has been achieved

If target cannot be reached

○ Stop scenario and save results

● Continue scenario without reaching goal

☑ Receive notification

Once the load of three hits per second has been reached, the Controller keeps running the scenario for an additional five minutes, adding or subtracting Vusers as needed to keep the actual measurement within 6% of the specified target. This is to ensure that the server can sustain the load for a period of time.

3 Do not use recorded think time.

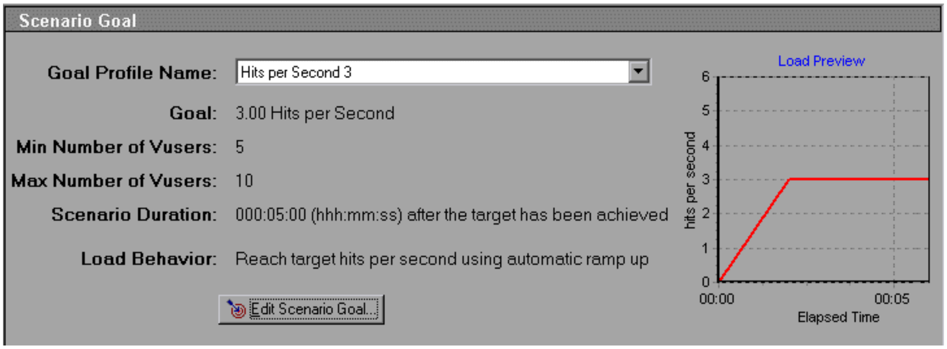
In the lower-left corner of the Edit Scenario Goal dialog box, ensure that **Do not change recorded think time** is cleared.

If you select this option, LoadRunner runs the scenario using the think time recorded in your script. You may need to increase the number of Vusers in your scenario in order to reach your target.

4 Close the Edit Scenario Goal dialog box.

Click **OK** to close the Edit Scenario Goal dialog box.

The scenario target information you entered appears in the Scenario Goal window.



Goal Profile Name: Hits per Second 3

Goal: 3.00 Hits per Second

Min Number of Vusers: 5

Max Number of Vusers: 10

Scenario Duration: 000:05:00 (hhh:mm:ss) after the target has been achieved

Load Behavior: Reach target hits per second using automatic ramp up

Which online graphs should I monitor?

After you have defined the test goal and load behavior, you are ready to configure the LoadRunner monitors. In this test, you should monitor the Hits per Second graph to follow the load that is generated on the server. You also want to monitor the Transaction Response Time graph to see what response times your customers will have when the server is under load. In addition, you can monitor the effect of load on the Throughput and Windows Resources graphs.

The Hits per Second, Transaction Response Time, and Throughput monitors have been configured for you. To configure the Windows Resources monitor, follow the procedures in Lesson 7, “Creating a Load Testing Scenario.”

How do I run a goal-oriented scenario?

Now that you have configured the scenario and goal settings, you are ready to start the test and monitor the application under load. In this section you will run your goal-oriented scenario and examine the test behavior.

1 Open the Controller window's Run tab.

Select the **Run** tab at the bottom of the screen.

Since the scenario has not run yet, all the counters remain at zero and the graphs are blank. When you start the scenario in the next step, the graphs and counters will begin to display information.

2 Specify a name for the Results directory.

Select **Results** > **Results Settings** to open the Set Results Directory dialog box, and enter a unique name for the result set, for example, travel_agent_3hps.

3 Start the scenario.

▶ Click the **Start Scenario** button or select **Scenario** > **Start**.

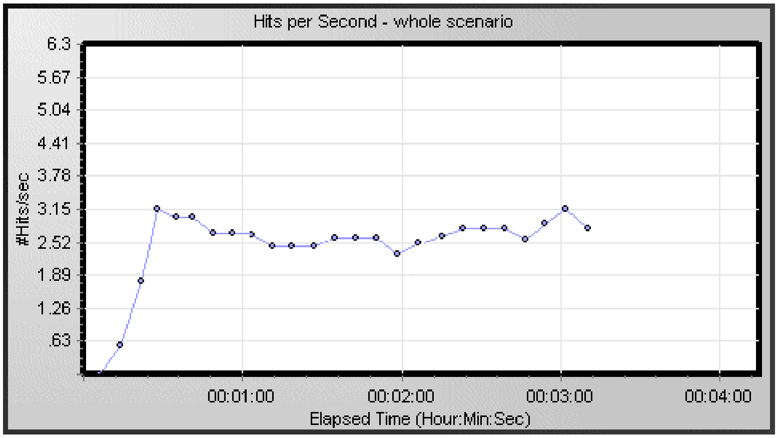
The Controller begins the scenario.

You will see that 5 Vusers are ramped up and start running, as LoadRunner attempts to generate the required goal of three hits per second.

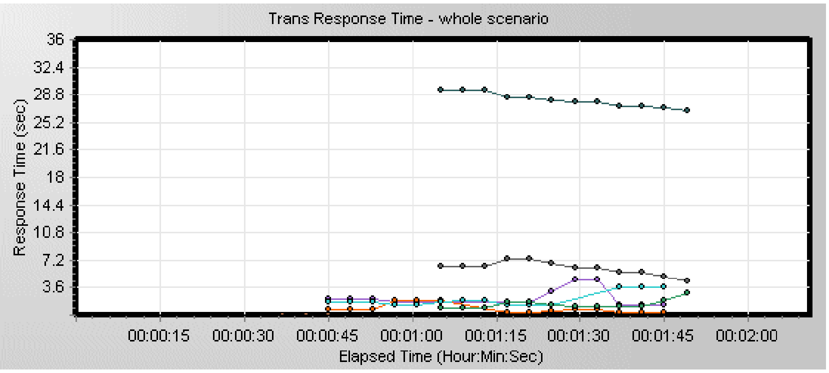
During the test, the Controller automatically starts and stops Vusers to maintain the given goal.

4 View the online graphs.

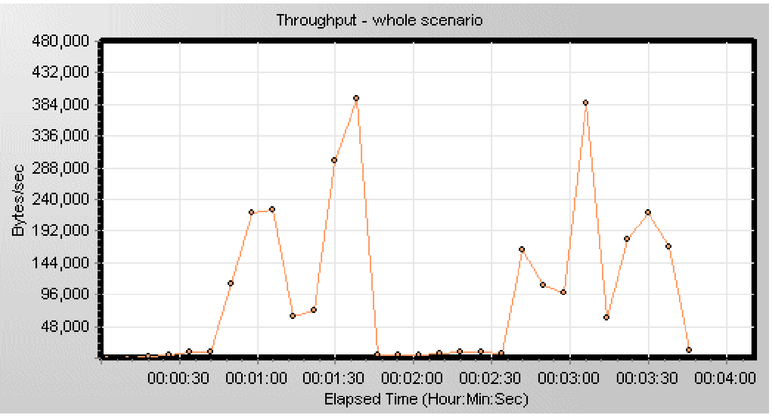
The Hits per Second graph shows the number of hits (HTTP requests) made to the Web server by Vusers during each second of the scenario run. You can see that after a short period the required level of load is reached.



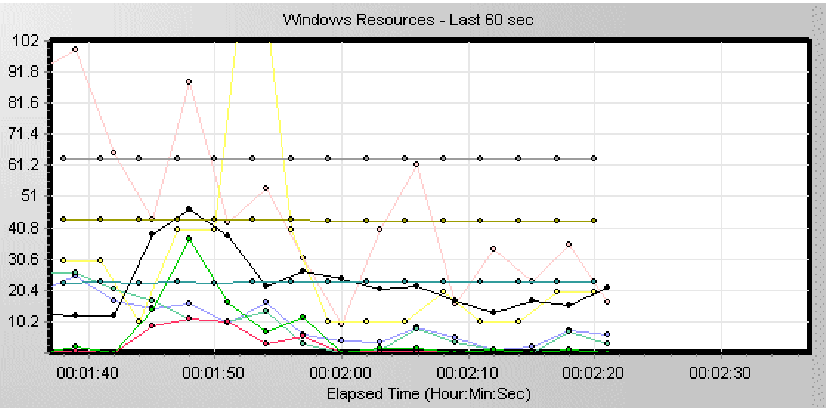
The Transaction Response Time graph shows the amount of time it takes for each transaction to be completed. It is very important to watch transaction response time to see the response times your customers will have when the server is under load.



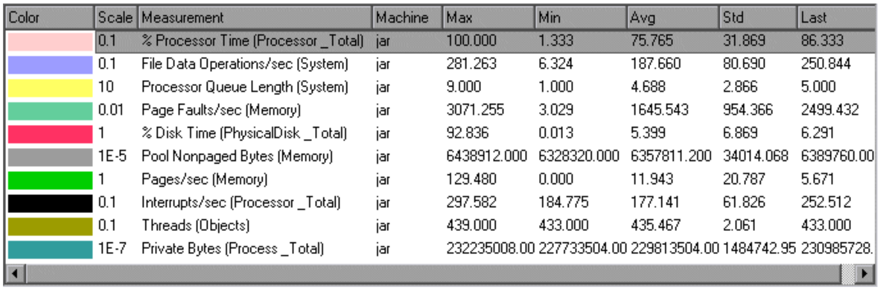
You can also look at the Throughput graph by selecting **Throughput** in the Available Graphs tree, and dragging it into the graph viewing area. This graph shows the amount of data that the Vusers receive from the Web server at each second.



You can monitor the server's Windows Resources usage for processor, disk, or memory utilization problems. Monitoring during a test can help you pinpoint the source of poor performance immediately.



You can check the Windows Resources graph legend for a list of measurements.



Color	Scale	Measurement	Machine	Max	Min	Avg	Std	Last
🟥	0.1	% Processor Time (Processor _Total)	jar	100.000	1.333	75.765	31.869	86.333
🟪	0.1	File Data Operations/sec (System)	jar	281.263	6.324	187.660	80.690	250.844
🟨	10	Processor Queue Length (System)	jar	9.000	1.000	4.688	2.866	5.000
🟩	0.01	Page Faults/sec (Memory)	jar	3071.255	3.029	1645.543	954.366	2499.432
🟥	1	% Disk Time (PhysicalDisk _Total)	jar	92.836	0.013	5.399	6.869	6.291
◼	1E-5	Pool Nonpaged Bytes (Memory)	jar	6438912.000	6328320.000	6357811.200	34014.068	6389760.00
🟩	1	Pages/sec (Memory)	jar	129.480	0.000	11.943	20.787	5.671
⬛	0.1	Interrupts/sec (Processor _Total)	jar	297.582	184.775	177.141	61.826	252.512
🟫	0.1	Threads (Objects)	jar	439.000	433.000	435.467	2.061	433.000
🟦	1E-7	Private Bytes (Process _Total)	jar	232235008.00	227733504.00	229813504.00	1484742.95	230985728.

Did I achieve my goal?

The objective of this lesson was to make sure that the system would provide an acceptable level of service to the customer under expected real-life workload. To emulate such conditions, you set a load target of three hits per second for the duration of the scenario, when running a minimum of five and maximum of ten Vusers. If three hits are made by Vusers on the server during each second of the scenario run when running between five and ten Vusers, then your goal parameters have been achieved. If your target of three hits per second is not reached, LoadRunner displays a message that the target you defined cannot be achieved.

Note: Since the license limits you to running a maximum of ten Vusers, your goal might not be reached.

After you have run the test, you should save the scenario settings for future use. To save the scenario, select **File** > **Save** or click the **Save** button and enter a scenario name in the File name box.

Where To Go From Here

Now that you have designed and run a goal-oriented scenario, you can proceed to Lesson 10, “Analyzing Your Scenario.”

10

Analyzing Your Scenario

In the previous lessons you learned how to design, control, and execute a scenario run. Once you have loaded your server, you want to analyze the run, and pinpoint the problems that need to be eliminated to improve system performance.

The graphs and reports produced during your analysis session present important information about the performance of your scenario. Using these graphs and reports, you can easily pinpoint and identify the bottlenecks in your application, and determine what changes need to be made to your system to improve its performance.

In this lesson you will cover the following topics:

- How does an analysis session work?
- How do I start my analysis session?
- The Analysis Window at a Glance
- Did I reach my goals? The Service Level Agreement
- How do I view a summary of performance?
- How do I graphically view performance?
- Did my server perform well?
- How can I pinpoint the source of the problem?
- What other information can I gather about my scenario run?
- How can I publish my findings?
- Conclusion

How does an analysis session work?

The aim of the analysis session is to find the failures in your system's performance and then pinpoint the source of these failures.

1. Were the test expectations met? What was the transaction response time on the user's end under load? Did the SLA meet or deviate from its goals? What was the average transaction response time of the transactions?
2. What parts of the system could have contributed to the decline in performance? What was the response time of the network and servers?
3. Can you find a possible cause by correlating the transaction times and backend monitor matrix?

In the following sections, you will learn how to open LoadRunner Analysis, and build and view graphs and reports that will help you find performance problems and pinpoint the sources of these problems.

How do I start my analysis session?

1 Open HP LoadRunner.

Select **Start** > **Programs** > **LoadRunner** > **LoadRunner**. HP LoadRunner Launcher window opens.

2 Open LoadRunner Analysis.

In the **Load Testing** tab, click **Analyze Load Tests**. HP LoadRunner Analysis opens.

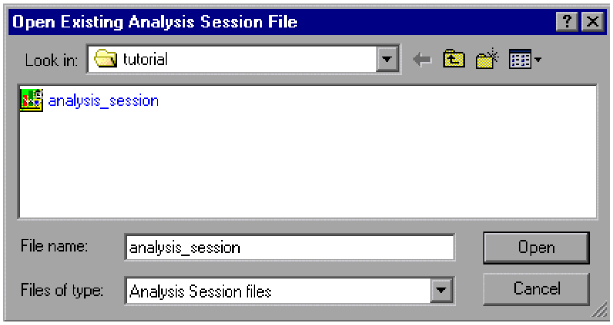
3 Open the analysis session file.

For the purpose of this section of the tutorial, and for more interesting results, we ran a test scenario similar to those you ran in the previous lessons. This time, however, the test incorporated 70 Vusers rather than 10 Vusers. You will now open the analysis session created from the results of this scenario.

In the Analysis window, select **File** > **Open**. The Open Existing Analysis Session File dialog box opens.

From the *<LoadRunner Installation>***Tutorial** folder, select **analysis_session** and click **Open**.

Note: If you are prompted to convert the session from an older version of LoadRunner to a newer version, click **OK**.



Open Existing Analysis Session File

Look in: tutorial

analysis_session

File name: analysis_session

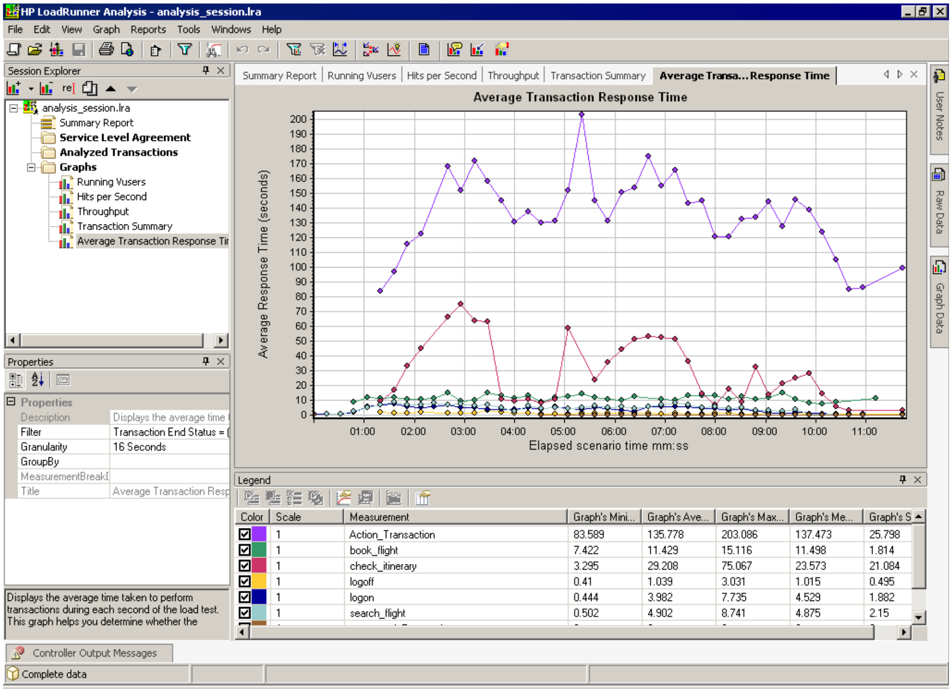
Files of type: Analysis Session files

Analysis opens the session file in the Analysis window.

The Analysis Window at a Glance

Analysis contains the following primary windows:

- Session Explorer
- Properties window
- Graph Viewing Area
- Graph Legend



Properties	
Description	Displays the average time
Filter	Transaction End Status = (
Granularity	16 Seconds
GroupBy	
MeasurementBreak[	
Title	Average Transaction Resp

Displays the average time taken to perform transactions during each second of the load test. This graph helps you determine whether the

Color	Scale	Measurement	Graph's Mini...	Graph's Ave...	Graph's Max...	Graph's Me...	Graph's S
☑ 🟪	1	Action_Transaction	83.589	135.778	203.086	137.473	25.798
☑ 🟩	1	book_flight	7.422	11.429	15.116	11.498	1.814
☑ 🟥	1	check_itinerary	3.295	29.208	75.067	23.573	21.084
☑ 🟨	1	logoff	0.41	1.039	3.031	1.015	0.495
☑ 🟦	1	logon	0.444	3.982	7.735	4.529	1.882
☑ 🟦	1	search_flight	0.502	4.902	8.741	4.875	2.15

- **Session Explorer pane.** In the upper left pane, Analysis shows the reports and graphs that are open for viewing. From here you can display new reports or graphs that do not appear when Analysis opens, or delete ones that you no longer want to view.
- **Properties window pane.** In the lower left pane, the Properties window displays the details of the graph or report you selected in the Session Explorer. Fields that appear in black are editable.

- **Graph Viewing Area.** In the upper right pane, Analysis displays the graphs. By default, the Summary Report is displayed in this area when you open a session.
- **Graph Legend.** In the lower right pane, you can view data from the selected graph.

Note: There are additional windows that can be accessed from the toolbar that provide additional information. These windows can be dragged and dropped anywhere on the screen.

Did I reach my goals? The Service Level Agreement

In this section, you will be introduced to the Service Level Agreement, or SLA.

SLAs are specific goals that you define for your load test scenario. Analysis compares these goals against performance related data that LoadRunner gathers and stores during the course of a run, and then determines the SLA status (**Pass** or **Fail**) for the goal.

For example, you can define a specific goal, or threshold, for the average transaction response time measurement for any number of transactions in your script.

After the test has run, LoadRunner compares the goals you defined against the actual recorded average transaction response times. Analysis displays the status of each defined SLA, either **Pass** or **Fail**. For example, if the actual average transaction response time did not exceed the threshold you defined, the SLA status will be **Pass**.

As part of your goal definition, you can instruct the SLA to take load criteria into account. This means that the acceptable threshold will vary depending on the level of load, for example, Running Vusers, Throughput, and so on. As the load increases, you can allow a higher threshold.

Depending on your defined goal, LoadRunner determines SLA statuses in one of the following ways:

- **SLA status determined at time intervals over a timeline.** Analysis displays SLA statuses at set time intervals (for example, every 5 seconds) over a timeline within the run.
- **SLA status determined over the whole run.** Analysis displays a single SLA status for the whole scenario run.

SLAs can be defined before running a scenario in the Controller, or after in Analysis itself.

In the section that follows, you will define an SLA using our HP Web Tours example. Assume that the administrator of HP Web Tours would like to know whenever the average transaction response times of the **book_flight** and **search_flight** transactions exceeds a certain value. To do this, you select the transactions and then set the threshold values. These threshold values are the maximum amounts of time that would be acceptable as average transaction response times.

You will also set these threshold values to take certain load criteria into account; in this case Running Vusers. In other words, as the number of Running Vusers increases, the threshold value rises.

The reason for this is that although the HP Web Tours administrator would like the average transaction response times to be as low as possible, it is understood that at certain times of the year it is reasonable to assume that the HP Web Tours site will have to handle a higher load than at other times of the year. For example, during peak travel season, a higher number of travel agents log on to the site to book flights, check itineraries, and so on. Given this understandably higher load, at these times a slightly longer average transaction response time will be acceptable.

You will set the SLA to take three load scenarios into account: light load, average load, and heavy load. Each scenario will have its own threshold value.

How do I define an SLA?

You will define an SLA in Analysis after the scenario run.

Note: The preferred place to define an SLA is before a scenario run in the Controller. However, for the purposes of this tutorial, because you are not analyzing the same test scenario that you ran in previous lessons, you will define the SLA in Analysis. To define an SLA in the Controller, click **New** in the Service Level Agreement section of the Design tab.

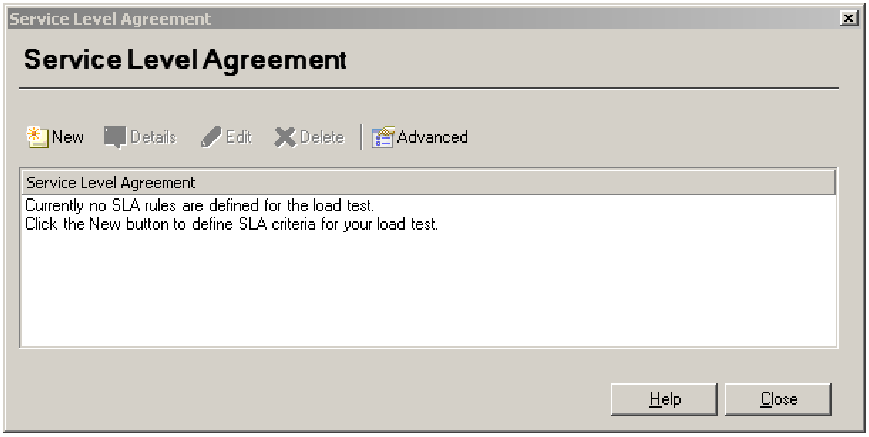
You will now define an SLA that will set specific goals for the average transaction response times for the **book_flight** and **search_flight** transactions in the sample session file.

The average transaction response times will be measured at set time intervals within the run.

To define an SLA:

1 **Open the SLA Configuration Wizard.**

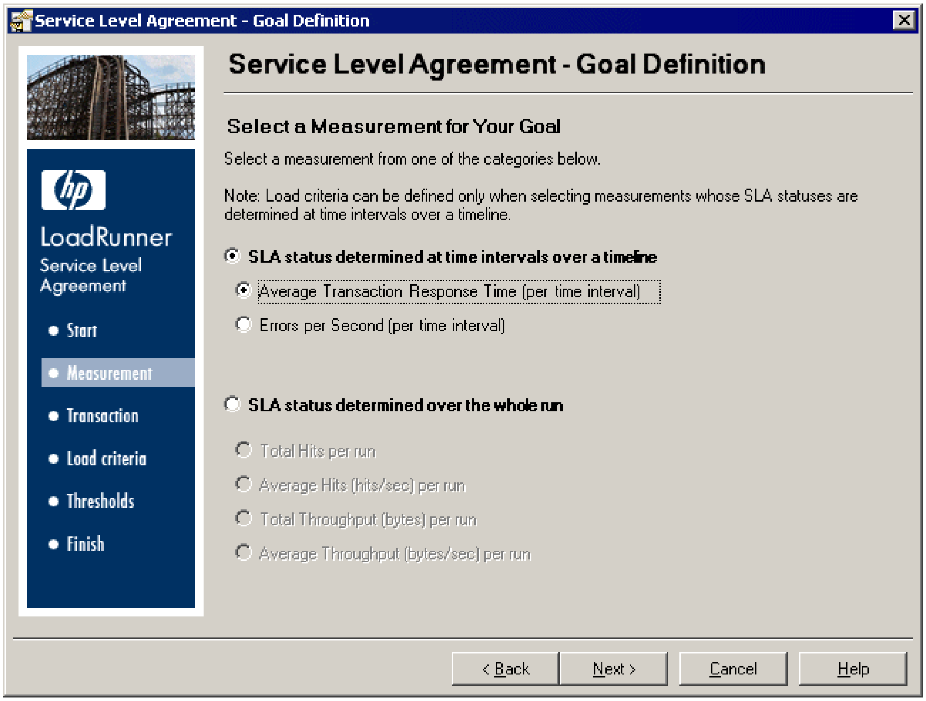
Select **Tools** > **Configure SLA Rules**. The Service Level Agreement dialog box opens. Click **New** to open the wizard.



Service Level Agreement
Currently no SLA rules are defined for the load test.
Click the New button to define SLA criteria for your load test.

1 Select a measurement for your goal.

Under **SLA status determined at time intervals over a timeline**, select **Average Transaction Response Time**. Click **Next**.



Service Level Agreement - Goal Definition

Service Level Agreement - Goal Definition

Select a Measurement for Your Goal

Select a measurement from one of the categories below.

Note: Load criteria can be defined only when selecting measurements whose SLA statuses are determined at time intervals over a timeline.

● **SLA status determined at time intervals over a timeline**

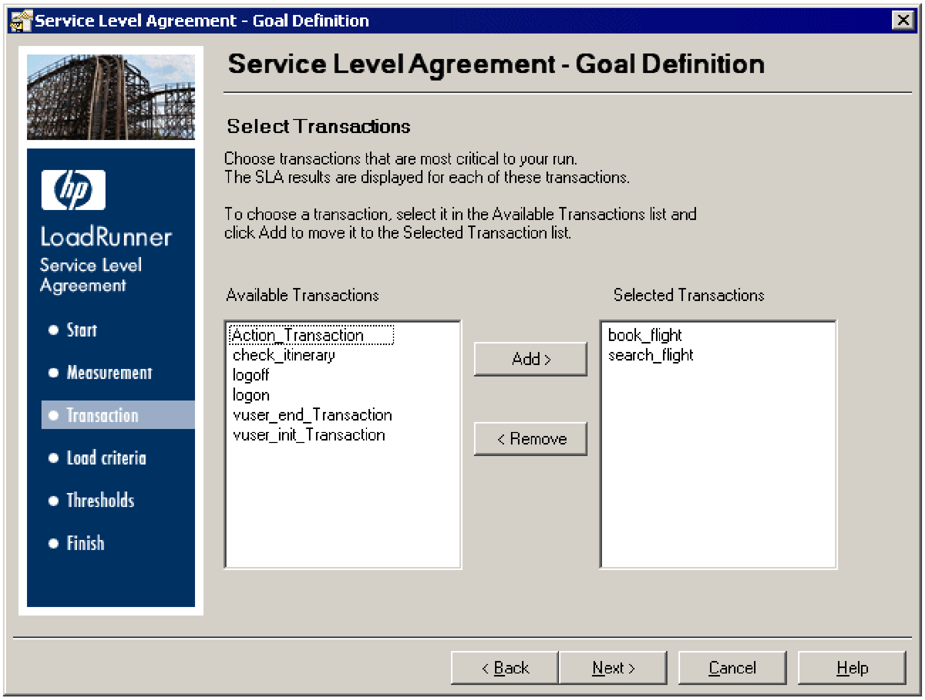
- ● Average Transaction Response Time (per time interval)
- ○ Errors per Second (per time interval)

○ **SLA status determined over the whole run**

- ○ Total Hits per run
- ○ Average Hits (hits/sec) per run
- ○ Total Throughput (bytes) per run
- ○ Average Throughput (bytes/sec) per run

2 Select transactions to monitor.

In this step, you select the transactions you would like to monitor from the **Available Transactions** list, which is a list of all the transactions in the script. Double-click the **book_flight** and **search_flight** transactions to select them. Click **Next**.



Service Level Agreement - Goal Definition

Service Level Agreement - Goal Definition

Select Transactions

Choose transactions that are most critical to your run.
The SLA results are displayed for each of these transactions.

To choose a transaction, select it in the Available Transactions list and
click Add to move it to the Selected Transaction list.

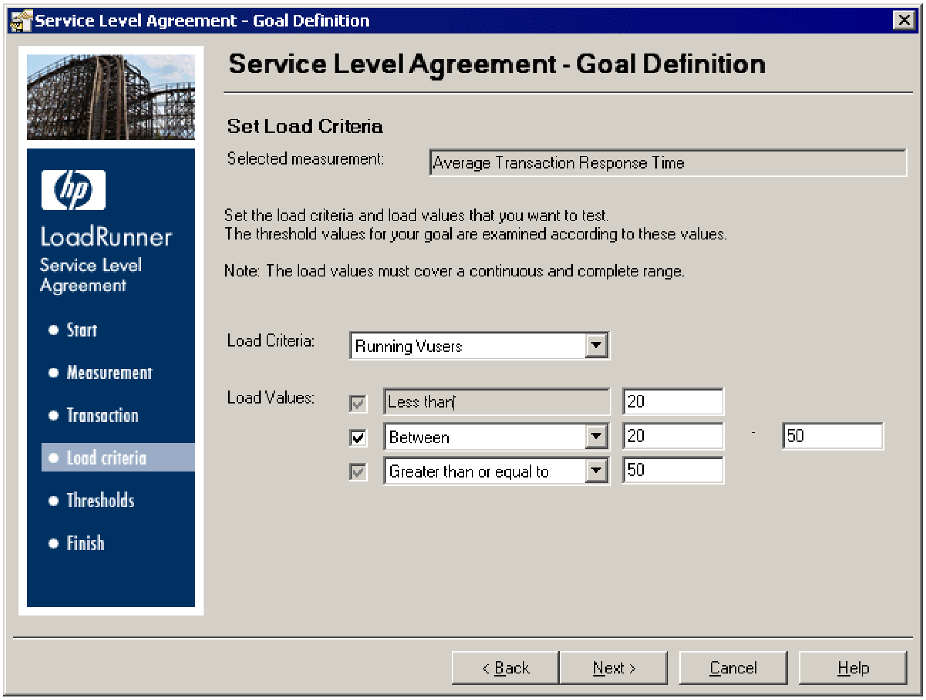
Available Transactions:
Action_Transaction
check_itinerary
logoff
logon
vuser_end_Transaction
vuser_init_Transaction

Selected Transactions:
book_flight
search_flight

3 Set the load criteria.

In this step, you instruct the SLA to take different load scenarios into account.

Select Running Vusers from the Load Criteria drop down list and set the Load Values to look as below:



Service Level Agreement - Goal Definition

Service Level Agreement - Goal Definition

Set Load Criteria

Selected measurement: Average Transaction Response Time

Set the load criteria and load values that you want to test.
The threshold values for your goal are examined according to these values.

Note: The load values must cover a continuous and complete range.

Load Criteria: Running Vusers

Load Values:

☑ Less than 20
☑ Between 20 - 50
☑ Greater than or equal to 50

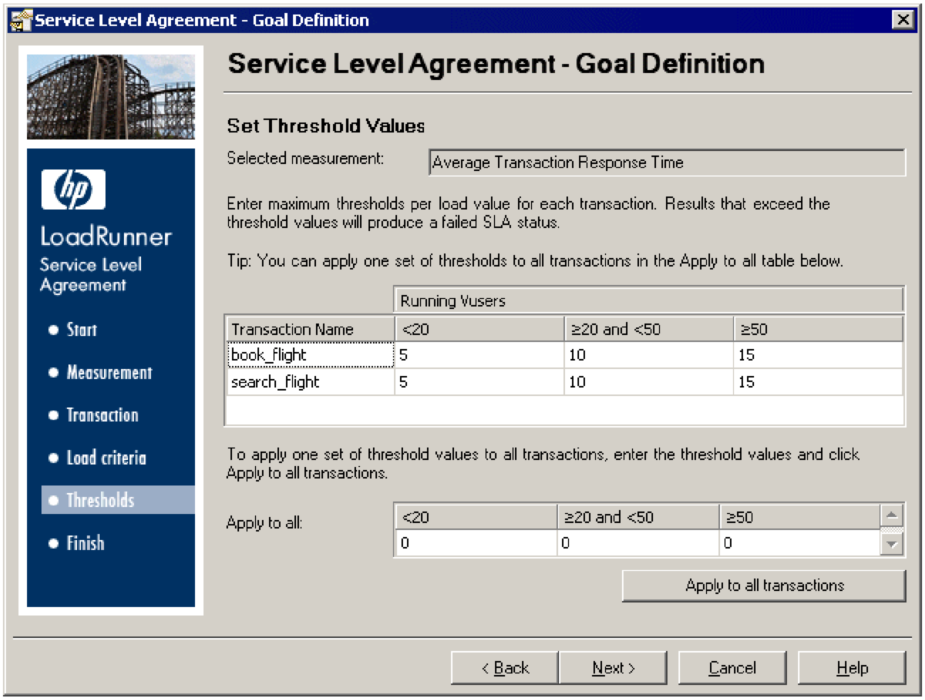
You have just set the SLA to determine an acceptable average transaction response time over three potential load scenarios:

- **Light load.** Between 0 and 19 Vusers
- **Average load.** Between 20 and 49 Vusers
- **Heavy load.** More than 50 Vusers

4 Set threshold values.

In this step, you will define the acceptable Average transaction response times for the **book_flight** and **search_flight** transactions.

Set the threshold values to look as below:



Service Level Agreement - Goal Definition

Set Threshold Values

Selected measurement: Average Transaction Response Time

Enter maximum thresholds per load value for each transaction. Results that exceed the threshold values will produce a failed SLA status.

Tip: You can apply one set of thresholds to all transactions in the Apply to all table below.

	Running Vusers		
Transaction Name	<20	≥20 and <50	≥50
book_flight	5	10	15
search_flight	5	10	15

To apply one set of threshold values to all transactions, enter the threshold values and click Apply to all transactions.

Apply to all:

<20	≥20 and <50	≥50
0	0	0

You have just determined that for both selected transactions, the following average transaction response times are acceptable:

- **Light load.** 5 seconds or less
- **Average load.** 10 seconds or less
- **Heavy load.** 15 seconds or less

Note: Threshold values for selected transactions do not have to be the same. You can assign different values for each transaction.

---

5 Save the SLA.

To save the SLA and close the wizard, click **Next** then **Finish** then **Close** on the pages that follow.

Analysis applies your SLA settings to the default Summary Report. The report is then updated to include all the relevant SLA information.

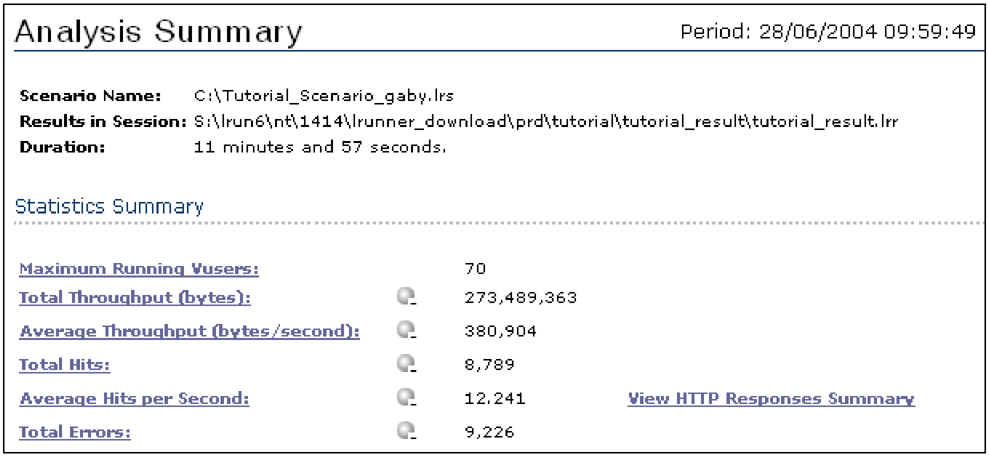
How do I view a summary of performance?

The Summary Report provides general information and statistics about the scenario run, as well as all relevant SLA information. For example, what were the worst performing transactions in terms of defined SLAs, how specific transactions performed over set time intervals, and overall SLA statuses. You open the Summary report from the Session Explorer.

The report contains the following sections:

What are the overall scenario statistics?

In the Statistics Summary section, you can see that a maximum of 70 Vusers ran in this test. Other statistics such as the total/average throughput, and the total/average hits are also logged here for your information.



Analysis Summary

Period: 28/06/2004 09:59:49

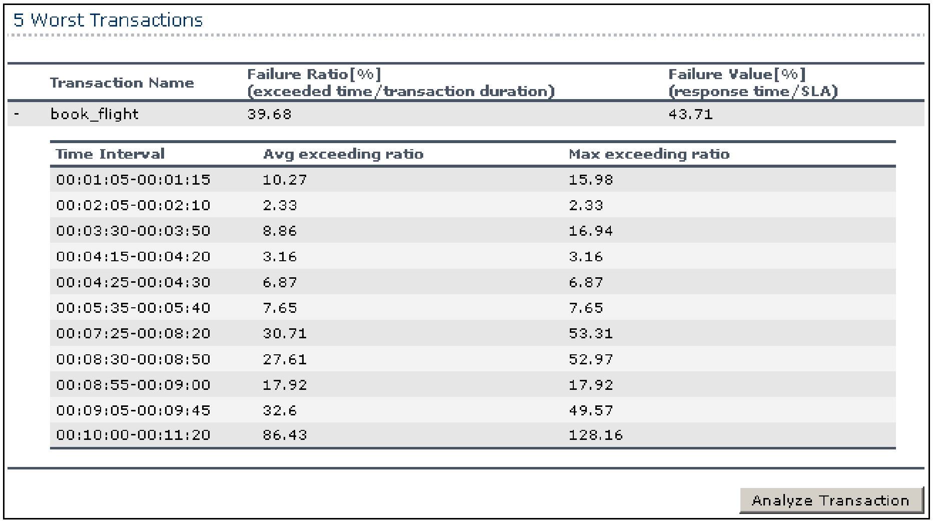
Scenario Name: C:\Tutorial_Scenario_gaby.lrs
Results in Session: S:\lrun6\nt\1414\lrunner_download\prd\tutorial\tutorial_result\tutorial_result.lrr
Duration: 11 minutes and 57 seconds.

Statistics Summary

Maximum Running Vusers:	70	
Total Throughput (bytes):	273,489,363	
Average Throughput (bytes/second):	380,904	
Total Hits:	8,789	
Average Hits per Second:	12.241	View HTTP Responses Summary
Total Errors:	9,226	

What were the worst performing transactions?

The 5 Worst Transactions table shows you up to the five worst performing transactions where SLAs were defined.



5 Worst Transactions

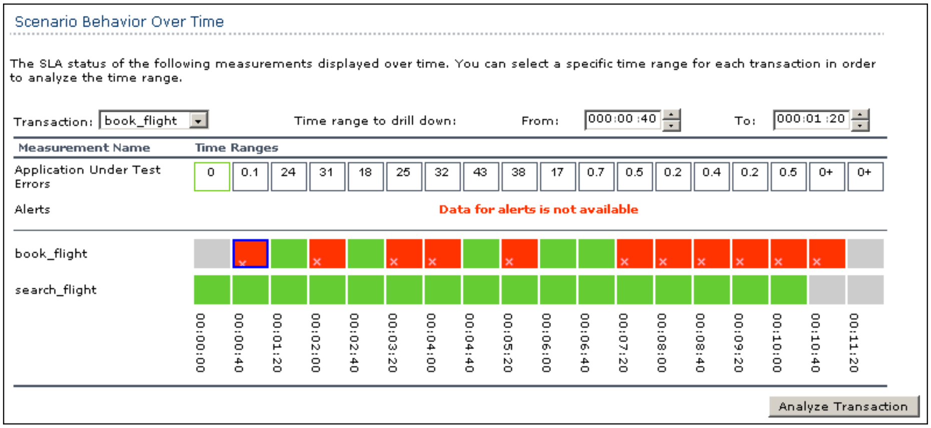
	Transaction Name	Failure Ratio[%] (exceeded time/transaction duration)	Failure Value[%] (response time/SLA)
-	book_flight	39.68	43.71

Time Interval	Avg exceeding ratio	Max exceeding ratio
00:01:05-00:01:15	10.27	15.98
00:02:05-00:02:10	2.33	2.33
00:03:30-00:03:50	8.86	16.94
00:04:15-00:04:20	3.16	3.16
00:04:25-00:04:30	6.87	6.87
00:05:35-00:05:40	7.65	7.65
00:07:25-00:08:20	30.71	53.31
00:08:30-00:08:50	27.61	52.97
00:08:55-00:09:00	17.92	17.92
00:09:05-00:09:45	32.6	49.57
00:10:00-00:11:20	86.43	128.16

You can see that over the duration of the **book_flight** transaction, the SLA threshold was exceeded 39.68% of the time. The average percentage by which it exceeded the SLA threshold over the whole run was 43.71%.

Over which time intervals was the SLA threshold exceeded?

The Scenario Behavior Over Time section shows how each transaction performed during different time intervals. The green squares show time intervals where the transaction performed within the SLA threshold, red squares where the transaction failed, and grey squares where no relevant SLA was defined.



Scenario Behavior Over Time

The SLA status of the following measurements displayed over time. You can select a specific time range for each transaction in order to analyze the time range.

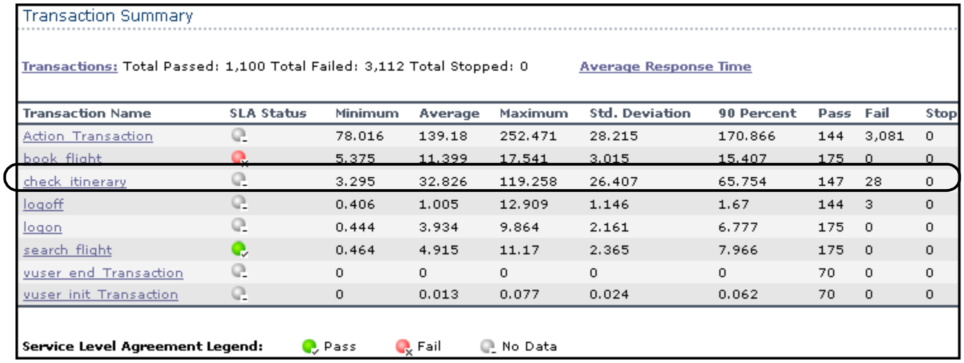
Transaction: book_flight   Time range to drill down: From: 000:00:40   To: 000:01:20

Measurement Name	Time Ranges																	
Application Under Test Errors	0	0.1	24	31	18	25	32	43	38	17	0.7	0.5	0.2	0.4	0.2	0.5	0+	0+
Alerts	Data for alerts is not available																	
book_flight	⬜	🟥	🟩	🟥	🟩	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	⬜
search_flight	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜
	00:00:00	00:00:40	00:01:20	00:02:00	00:02:40	00:03:20	00:04:00	00:04:40	00:05:20	00:06:00	00:06:40	00:07:20	00:08:00	00:08:40	00:09:20	00:10:00	00:10:40	00:11:20

You can see that of the two transactions where you defined an SLA, **search_flight** was within the threshold over all the measured time intervals, while **book_flight** exceeded the threshold in several intervals.

What was the overall transaction performance?

The Transaction Summary lists a summary of the behavior of each transaction.



Transaction Summary

Transactions: Total Passed: 1,100 Total Failed: 3,112 Total Stopped: 0   Average Response Time

Transaction Name	SLA Status	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action_Transaction	⚪	78.016	139.18	252.471	28.215	170.866	144	3,081	0
book_flight	🔴	5.375	11.399	17.541	3.015	15.407	175	0	0
check_itinerary	⚪	3.295	32.826	119.258	26.407	65.754	147	28	0
logoff	⚪	0.406	1.005	12.909	1.146	1.67	144	3	0
logon	⚪	0.444	3.934	9.864	2.161	6.777	175	0	0
search_flight	🟢	0.464	4.915	11.17	2.365	7.966	175	0	0
vuser_end_Transaction	⚪	0	0	0	0	0	70	0	0
vuser_init_Transaction	⚪	0	0.013	0.077	0.024	0.062	70	0	0

Service Level Agreement Legend: 🟢 Pass 🔴 Fail ⚪ No Data

Look at the response times of each transaction. The 90 Percent column displays the response time of 90% of the executions of a particular transaction. You can see that 90% of the **check_itinerary** transactions that were performed during the test run had a response time of 65.754 seconds. This is double its average response time, 32.826, which means that the majority of occurrences of this transaction had a very high response time.

We also see that this transaction failed 28 times.

Note how the SLA Status column shows the relevant overall SLA statuses: **Fail** for **book_flight** and **Pass** for **search flight**.

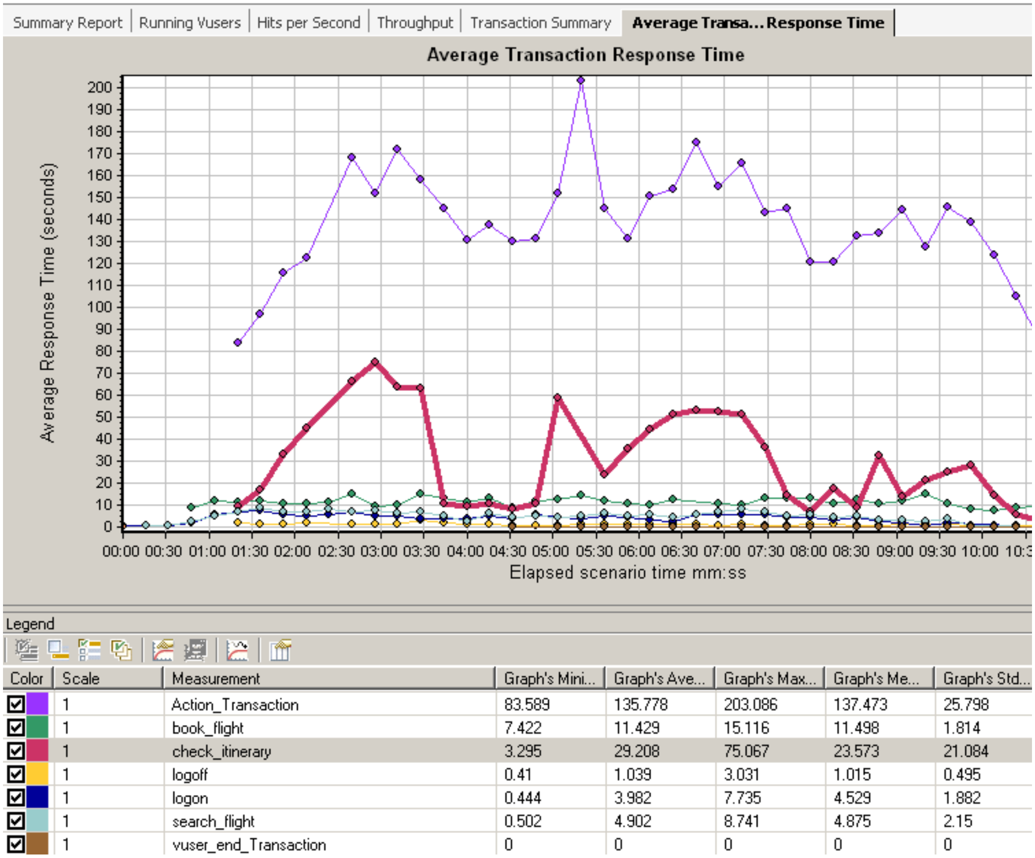
How do I graphically view performance?

You can access available graphs from the Session Explorer pane. You will now view and analyze the Average Transaction Response Time graph.

1 Open the Average Transaction Response Time graph.

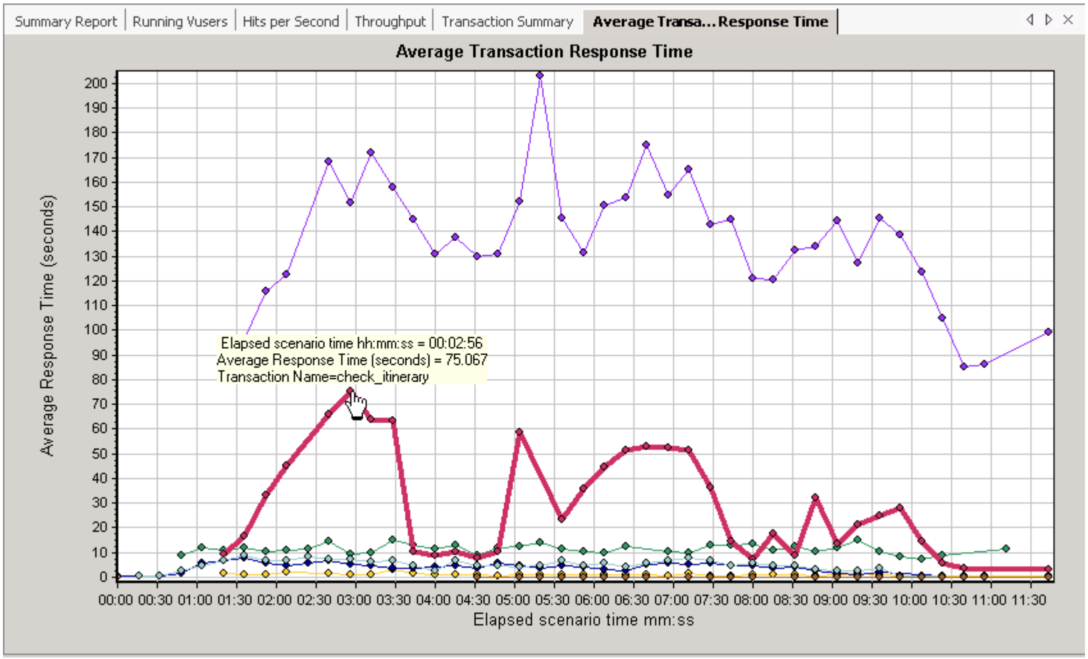
In the Session Explorer under **Graphs**, click **Average Transaction Response Time**. The Average Transaction Response Time graph opens in the graph viewing area.

In the Graph Legend, click the **check_itinerary** transaction. The check_itinerary transaction is highlighted in the graph and in the legend below the graph.



Color	Scale	Measurement	Graph's Mini...	Graph's Ave...	Graph's Max...	Graph's Me...	Graph's Std...
☑ 🟪	1	Action_Transaction	83.589	135.778	203.086	137.473	25.798
☑ 🟩	1	book_flight	7.422	11.429	15.116	11.498	1.814
☑ 🟥	1	check_itinerary	3.295	29.208	75.067	23.573	21.084
☑ 🟨	1	logoff	0.41	1.039	3.031	1.015	0.495
☑ 🟦	1	logon	0.444	3.982	7.735	4.529	1.882
☑ 🟦	1	search_flight	0.502	4.902	8.741	4.875	2.15
☑ 🟫	1	vuser_end_Transaction	0	0	0	0	0

The points on the graph represent the average response time of a transaction at a specific time during the scenario. Hold your cursor over a point in the graph. A yellow box appears, and displays the coordinates of that point.



Elapsed scenario time hh:mm:ss = 00:02:56
Average Response Time (seconds) = 75.067
Transaction Name=check_itinerary

2 Analyze the results.

Note how the average response time of the check_itinerary transaction fluctuates greatly, and reaches a peak of 75.067 seconds, 2:56 min. into the scenario run.

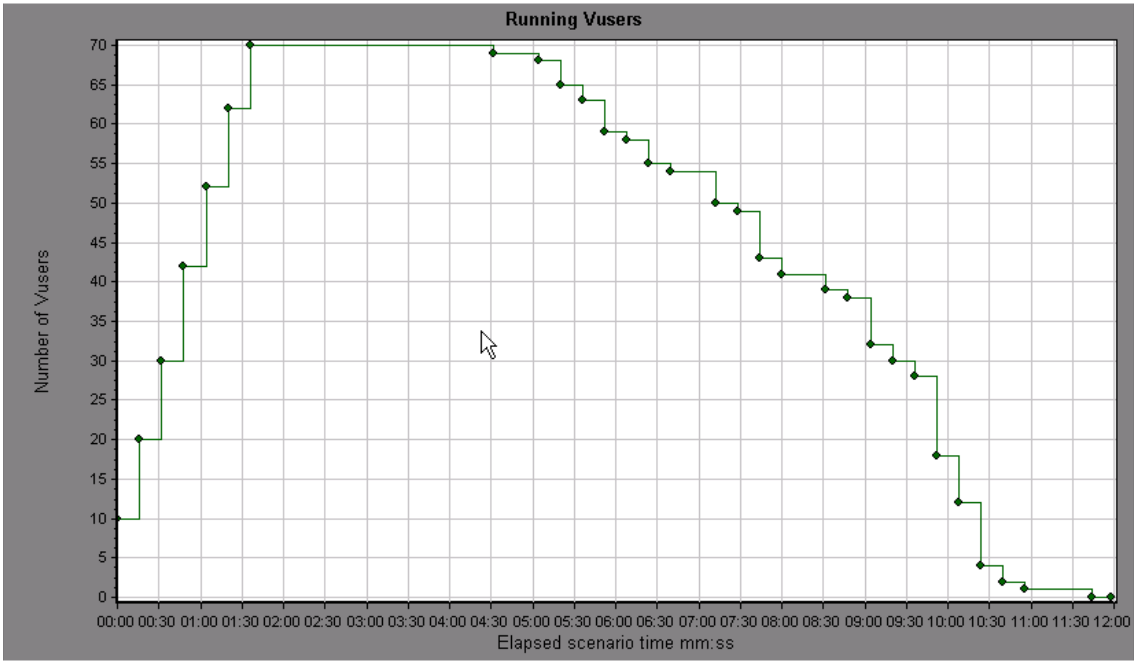
On a well-performing server, the transactions would follow a relatively stable average response time. At the bottom of the graph, note how the logon, logoff, book_flight, and search_flight transactions follow a more or less stable average response time.

Did my server perform well?

In the previous section you saw instability in your server's performance. Now you will analyze the effect of 70 running Vusers on the system's performance.

1 Study the behavior of the Vusers.

Click **Running Vusers** in the graph tree.



The Running Vusers graph opens in the graph viewing area. You can see that there was a gradual start of running Vusers at the beginning of the scenario run. Then, for a period of 3 minutes, 70 Vusers ran simultaneously, after which the Vusers gradually stopped running.

2 Filter the graph so that you see only the time slice when all the Vusers ran simultaneously.

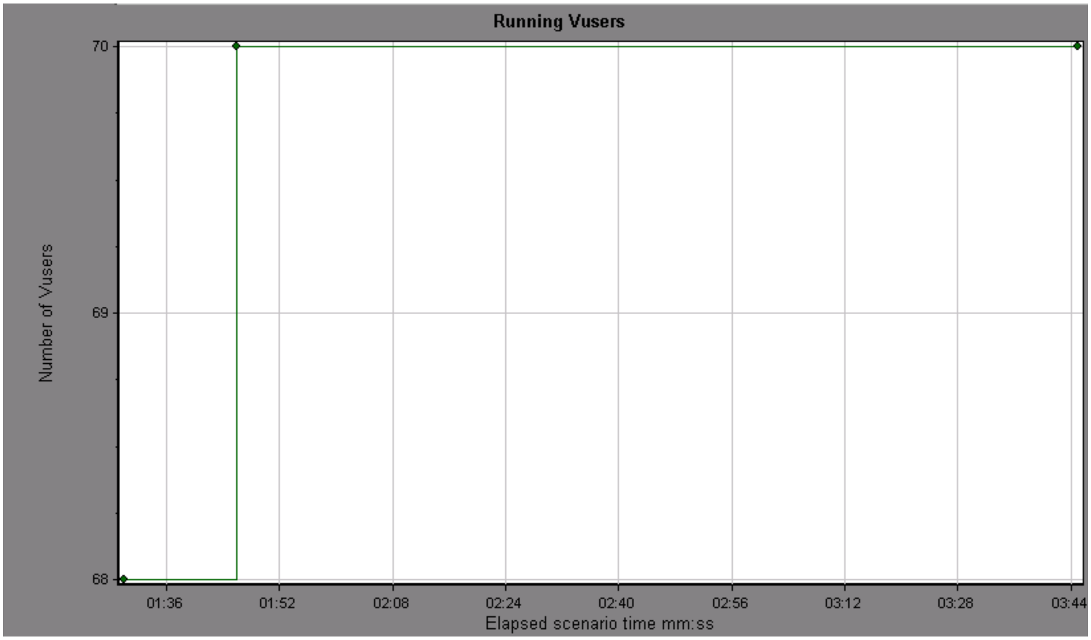
When you filter a graph, the graph data is narrowed down so that only the data for the condition that you specified is displayed. All other data is hidden.

Right-click the graph and select **Set Filter/Group By**, or alternatively, click the **Set Filter/Group By...** icon on the toolbar.

In the **Filter Condition** area, select the **Values** column of the **Scenario Elapsed Time** row. Click the down-arrow and select a time range from, 1:30 minutes to 3:45 minutes. Click **OK**.

In the Graph Settings dialog box, click **OK**.

The Running Vusers graph now displays only those Vusers running between 1:30 minutes and 3:45 minutes of the scenario run. All other Vusers have been filtered out.



Note: To clear the filter, you right-click the graph and select **Clear Filter/Group By**, or alternatively, click the **Clear Filter and Group By** icon on the toolbar.

3 Correlate the Running Vusers and Average Transaction Response Time graphs to compare their data.

You can join two graphs together to see the effect of one graph's data upon the other graph's data. This is called *correlating two graphs*.

For example, you can correlate the Running Vusers graph with the Average Transaction Response Time graph to see the effect of a large number of Vusers on the average response time of the transactions.

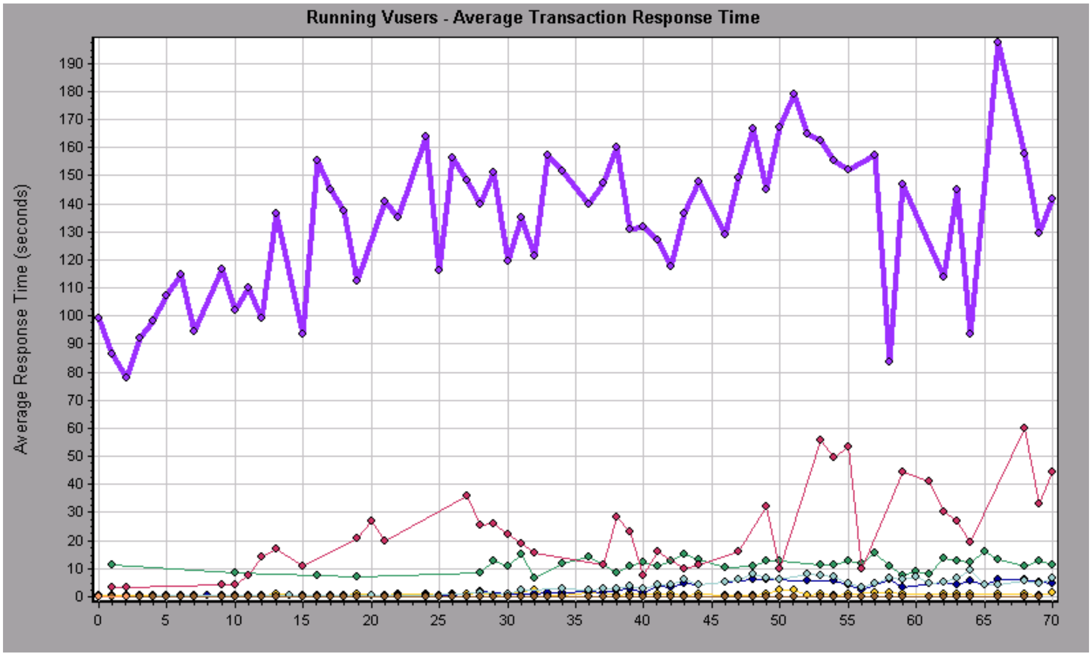
Right-click the Running Vusers graph and select **Clear Filter/Group By**.

Right-click the graph and select **Merge Graphs**.

From the **Select graph to merge with** list, select **Average Transaction Response Time**.

In the **Select type of merge** area, select **Correlate**, and click **OK**.

The Running Vusers and Average Transaction Response Time graphs are now represented by one graph, the Running Vusers - Average Transaction Response Time graph, that appears in the graph viewing area.



4 Analyze the correlated graph.

In this graph you can see that as the number of Vusers increases, the average response time of the check_itinerary transaction very gradually increases. In other words, the average response time steadily increases as the load increases.

At 64 Vusers, there is a sudden, sharp increase in the average response time. We say that the test **broke the server**. The response time clearly began to degrade when there were more than 64 Vusers running simultaneously.

Saving a template

Up to now you have filtered a graph and correlated two graphs. The next time you analyze a scenario, you might want to view the same graphs, with the same filter and merge conditions applied. You can save your merge and filter settings into a template, and apply them in another analysis session.

To save your template:

1. From the **Tools** menu, select **Templates** > **Save as Template...**
2. Enter an appropriate name for the template.
3. Clear the **Automatically apply this template to a new session** option.
4. Click **OK**.

Next time you open a new analysis session and want to use a saved template:

1. From the **Tools** menu, select **Templates** > **Apply/Edit Template.**
2. Select your template from the list, and click **Apply Template**.

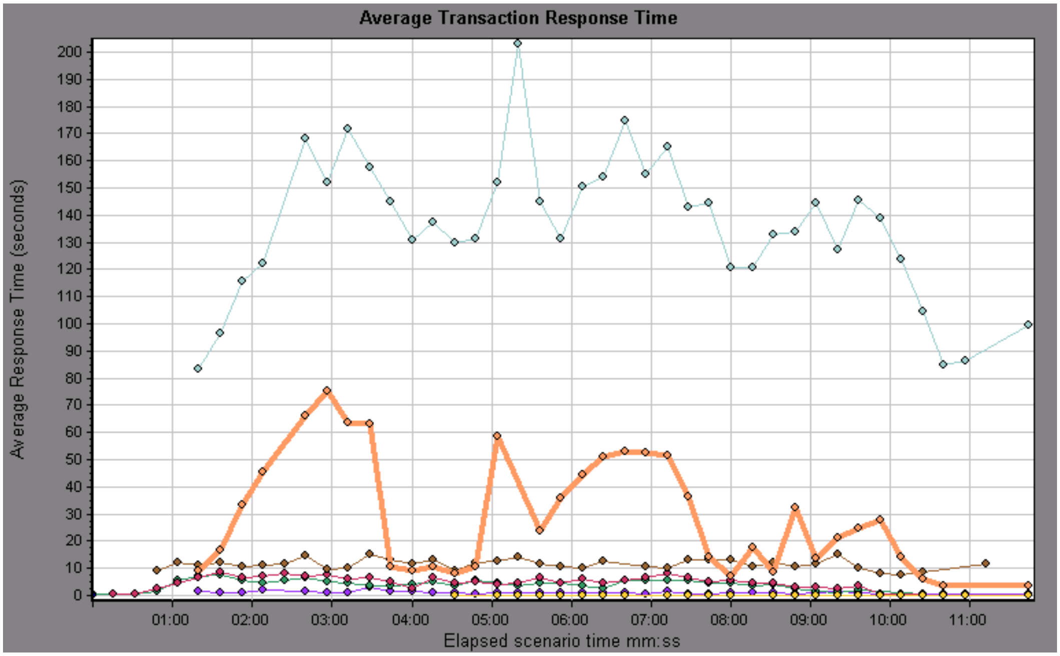
How can I pinpoint the source of the problem?

Until now, you have seen that an increase in load on the server had a negative impact on the average response time of the check_itinerary transaction.

You can drill down further into the check_itinerary transaction to see what system resources could have negatively influenced its performance.

Unique to LoadRunner Analysis, the Auto-correlate tool can merge all the graphs that contain data that could have had an effect on the response time of the check_itinerary transaction, and pinpoint what was happening at the moment the problem occurred.

1 From the graph tree, select the Average Transaction Response Time graph.



Look at the check_itinerary transaction, particularly at the slice of elapsed time between 1 and 4 minutes. The average response time started to increase almost immediately, until it peaked at nearly 3 minutes.

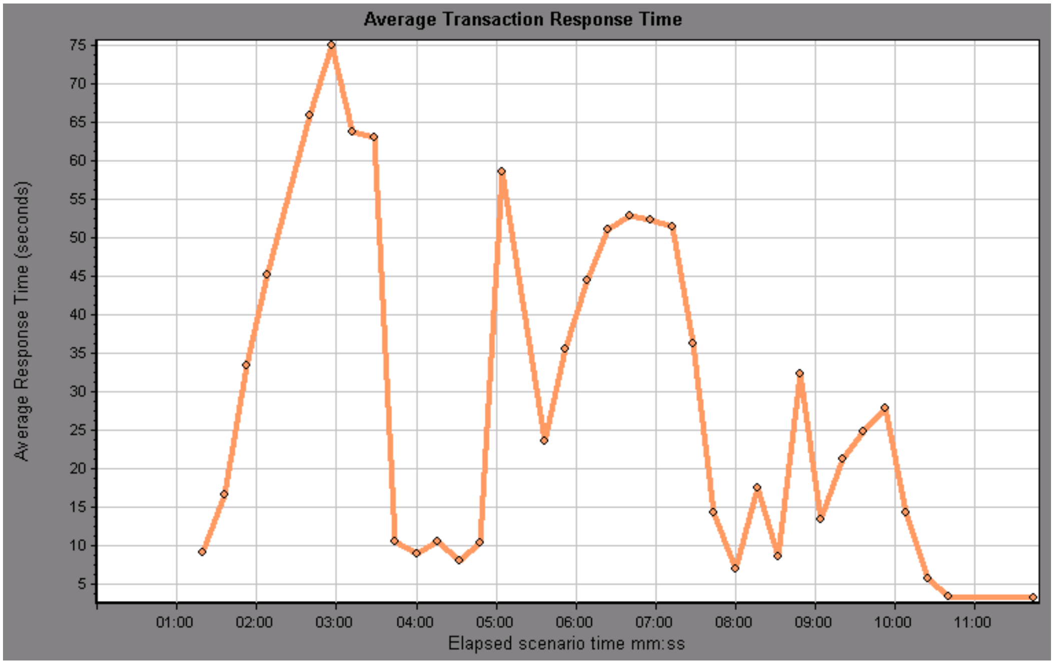
2 Filter the Average Transaction Response Time graph to display only the check_itinerary transaction.

Right-click the graph, and select **Set Filter/Group by**.

In the Transaction Name value box, select **check_itinerary**.

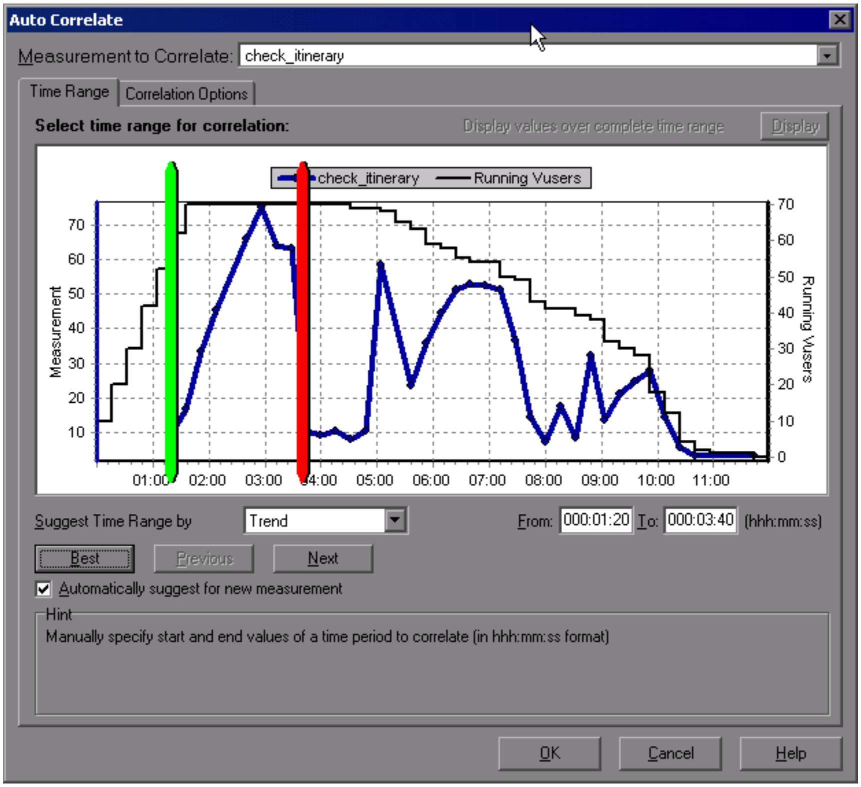
Click **OK**.

The filtered graph displays only the check_itinerary transaction and hides all the other transactions.



3 Auto-correlate the graph.

Right-click the graph, and select **Auto Correlate**.



Auto Correlate

Measurement to Correlate: check_itinerary

Select time range for correlation:

Suggest Time Range by: Trend

From: 000:01:20 To: 000:03:40 (hhh:mm:ss)

☑ Automatically suggest for new measurement

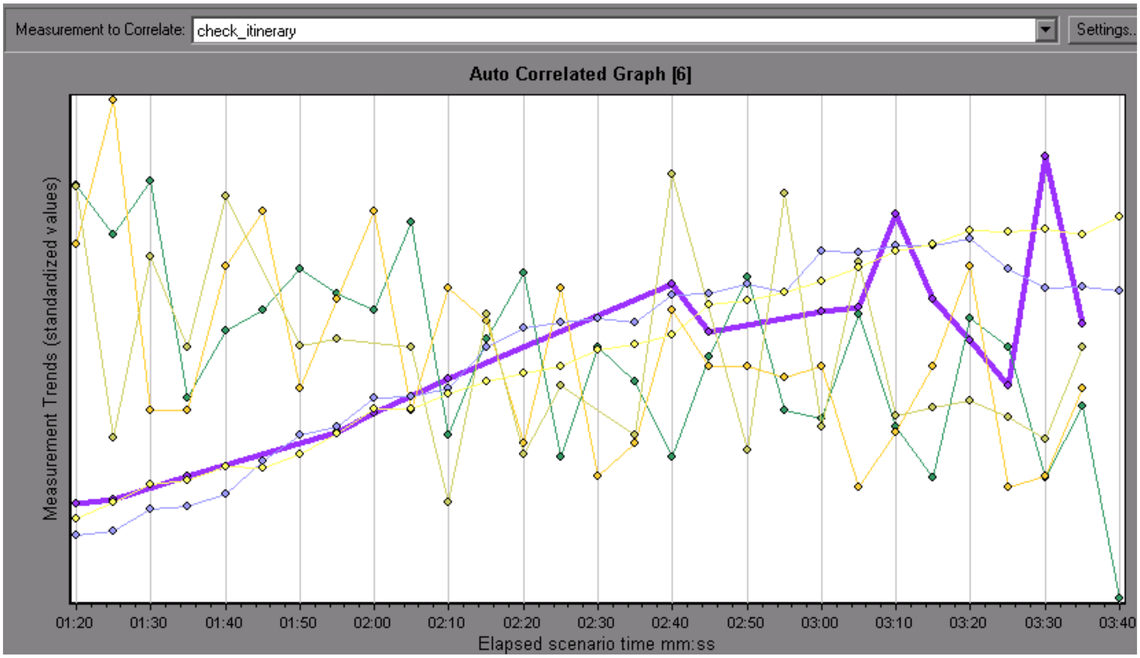
Hint

Manually specify start and end values of a time period to correlate (in hhh:mm:ss format)

In the Auto Correlate dialog box, make sure that the measurement to correlate is **check_itinerary**, and set the time range from **1:20** to **3:40** minutes **-** either by entering the times in the boxes, or by dragging the green and red poles into place along the Elapsed Scenario Time axis.

Click **OK**.

The auto-correlated graph opens in the graph viewing area. The check_itinerary transaction is highlighted.



Measurement to Correlate: check_itinerary

Settings...
Auto Correlated Graph [6]
Measurement Trends (standardized values)
01:20
01:30
01:40
01:50
02:00
02:10
02:20
02:30
02:40
02:50
03:00
03:10
03:20
03:30
03:40
Elapsed scenario time mm:ss

The auto-correlated graph is given a default name, **Auto Correlated Graph** [*number*].

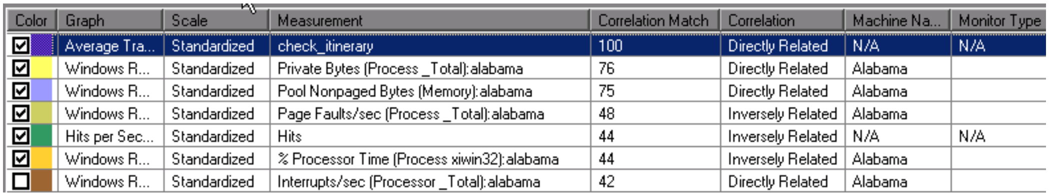
4 Rename the graph.

In the graph tree, right-click the **Auto Correlated Graph** [*number*] graph, and select **Rename Item**. The graph name becomes editable.

Type **Auto Correlated - check_itinerary** and press ENTER, or click anywhere in the Analysis window.

5 Analyze the auto-correlated graph.

Look at the legend below the graph.



Color	Graph	Scale	Measurement	Correlation Match	Correlation	Machine Na...	Monitor Type
☑ 🟪	Average Tra...	Standardized	check_itinerary	100	Directly Related	N/A	N/A
☑ 🟨	Windows R...	Standardized	Private Bytes (Process _Total):alabama	76	Directly Related	Alabama	
☑ 🟦	Windows R...	Standardized	Pool Nonpaged Bytes (Memory):alabama	75	Directly Related	Alabama	
☑ 🟩	Windows R...	Standardized	Page Faults/sec (Process _Total):alabama	48	Inversely Related	Alabama	
☑ 🟩	Hits per Sec...	Standardized	Hits	44	Inversely Related	N/A	N/A
☑ 🟨	Windows R...	Standardized	% Processor Time (Process xiwin32):alabama	44	Inversely Related	Alabama	
□ 🟫	Windows R...	Standardized	Interrupts/sec (Processor _Total):alabama	42	Directly Related	Alabama	

In the Measurement column you can see that the **Private Bytes** and **Pool Nonpaged Bytes**, both of which are memory-related measurements, have a Correlation Match of over 70% with the check_itinerary transaction. This means that the behavior of these elements was closely related to the behavior of the check_itinerary transaction during the specified time interval.

We can deduce that at the instant when the check_itinerary transaction's response time peaked, there was a shortage of system memory resources.

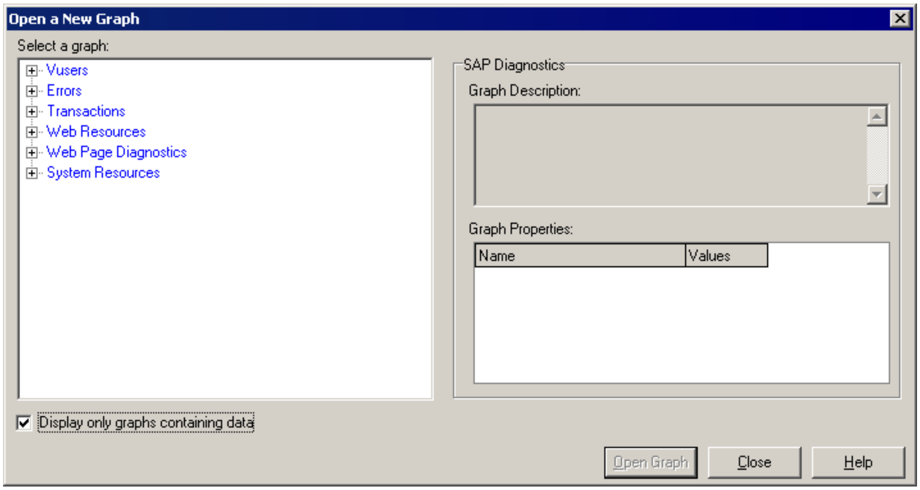
What other information can I gather about my scenario run?

In addition to the graphs that appear in the graph tree at the start of an analysis session, you can display different graphs to get other information about your scenario run.

1 Display a new graph.

Click the **Add New Graph** button on the toolbar or, alternatively, click **New Graph** in the graph tree.

The Open a New Graph dialog box opens and lists the categories of graphs that contain data and can be displayed.



- **Vusers.** Displays information about the Vusers and their status.
- **Errors.** Displays error statistics.
- **Transactions.** Displays data about transactions and their response times.
- **Web Resources.** Displays hits, throughput, and connection data.
- **Web Page Diagnostics** graphs display data about each monitored Web page in your script.
- **System Resource** graphs display system resource usage data.

In the Open a New Graph dialog box, click the "+" next to a category to expand it.

Select a graph and click **Open Graph**.

Click **Close** to close the Open a New Graph dialog box.

Now open several additional graphs to understand more about your scenario run.

How can I publish my findings?

You can publish the findings from your analysis session in an HTML or Microsoft Word report. The report is created using a designer template, and includes explanations and legends of the presented graphs and data.

HTML Reports

The HTML report can be opened and viewed in any browser.

To create an HTML report:

1 From the **Reports** menu, select **HTML Report...**

2 Select a file name for your report, and the path where you want to save it. Click **Save**.

Analysis creates the report and displays it in your Web browser. Note how the layout of the HTML report is very similar to the layout of your analysis session. You can click on the links in the left pane to see the various graphs. A description of each graph is given at the bottom of the page.

Microsoft Word Reports

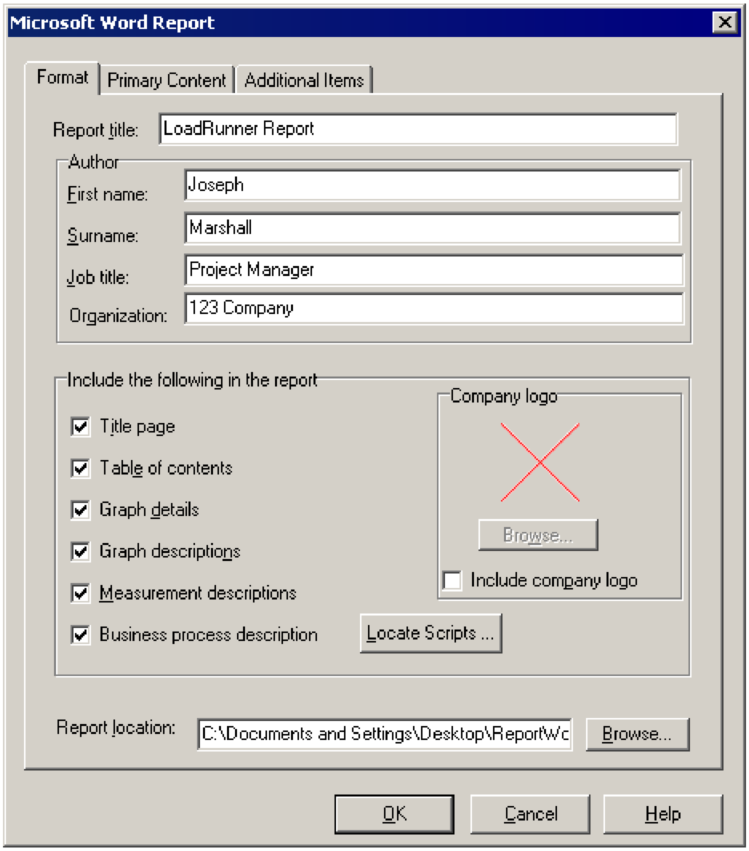
You can present your analysis session in a Microsoft Word report. The Word report is more comprehensive than the HTML report, because you have the option to include general information about the scenario, measurement descriptions, and so on. You can also format the report to include your company's name and logo, and the author's details.

Like any Microsoft Word file, the report is editable, so you can add further comments and findings after you have built the report.

To create a Microsoft Word report:

1 From the **Reports** menu, select **Microsoft Word Report**.

The Microsoft Word Report dialog box opens.



Report title: LoadRunner Report

First name: Joseph

Surname: Marshall

Job title: Project Manager

Organization: 123 Company

☑ Title page

☑ Table of contents

☑ Graph details

☑ Graph descriptions

☑ Measurement descriptions

☑ Business process description

☐ Include company logo

Report location: C:\Documents and Settings\Desktop\ReportWc

2 Click the **Format** tab:

➤ Enter a title for your report.

➤ Enter the author’s name, job title, and the company’s name.

➤ By default, the report will be built with a title page, table of contents, graph details and descriptions, and measurement descriptions. You can select options which add script details into the report, allowing you to view thumbnail images of the business process steps.

- ➤ In the **Company logo** area, select **Include company logo**, and click **Browse** to find the logo file path. Note that this must be a **.bmp** file.
- ➤ Enter a location to save your report.

3 Click the **Primary Content** tab.

- ➤ Select which sections of your scenario run and analysis session you want to include in your report. By default, all sections listed, except for the **Server Performance**, are selected.

  Select **Server performance**.

- ➤ Click **Edit**. The Executive Summary dialog box opens, where you type your objectives and conclusions.
  - ➤ In the **Objectives** box, type The objectives of the test scenario were to....
  - ➤ In the **Conclusions** box, type The conclusions I reached are as follows:

  Click **OK** to close the Executive Summary dialog box.

4 Click the **Additional Items** tab.

- ➤ Specify which graphs you want to include in the report. By default, all the graphs in your session are listed and selected, and the graphs notes will be included in your report.
- ➤ For more information, you can add graphs that you did not already open in your analysis session.

  Click the **Add** button. The Open a New Graph dialog box opens. Expand the System Resources category and select the **Windows Resources** graph. Click **Add Graph.** Click **Close** to close the Open a New Graph dialog box. The Windows Resources graph appears in the list of graphs that will be included in the report.

- ➤ You can specify the order in which to display the graphs in the report.

  Click **Average Transaction Response Time** to select the graph.

  Click the **Up** button until the graph appears under **Running Vusers**.

  In the report, the Average Transaction Response Time graph will follow the Running Vusers graph.

5 In the Microsoft Word Report dialog box, click **OK.**

The data is gathered and the report is created in a Microsoft Word file, which opens in Microsoft Word.

In addition to the graphs that you generated during your analysis session, the report includes an objective and a conclusion, and other sections and graphs that you chose to include while building the report.

Conclusion

In this lesson you learned the basics of defining a Service Level Agreement, analyzing a scenario run, and publishing your results in a report.

You have learned that performance problems can be pinpointed by studying various graphs that show bottlenecks in the server, possibly due to too heavy a load, and that you can pinpoint the sources of these bottlenecks by configuring graphs to display correlated data.