LoadRunner Tuning Module

Console User's Guide

Version 7.8

LoadRunner 7.8 Tuning Module Console User's Guide

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089 USA
Tel: (408) 822-5200
Toll Free: (800) TEST-911, (866) TOPAZ-4U
Fax: (408) 822-5300

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

LRTMCONUG7.8/01

Table of Contents

Welcome to LoadRunner Tuning Module

Welcome to LoadRunner Tuning Module, the proactive solution for optimizing production systems. LoadRunner Tuning Module's system-wide approach to optimization is a product of Mercury Interactive's expert knowledge and tuning methodologies.

LoadRunner Tuning Module's chief purpose is to enable the user to explore the network, detect bottlenecks, and assist during the tuning phase to enhance performance. LoadRunner Tuning Module combines the network topology, predefined tuning sessions and goals into a set of tests that pinpoint problematic components in the client network. In addition, LoadRunner Tuning Module helps the more advanced user to explore and tune the business processes by providing a simple and organized methodology.

Online Resources

LoadRunner Tuning Module includes the following online tools:

Read Me First provides last-minute news and information about LoadRunner Tuning Module.

Books Online displays the complete documentation set in PDF format. Online books can be read and printed using Adobe Acrobat Reader, which is included in the installation package. Check Mercury Interactive's Customer Support Web site for updates to LoadRunner Tuning Module online books.

LoadRunner Tuning Module **Function Reference** gives you online access to all of LoadRunner Tuning Module's functions that you can use when creating Vuser scripts, including examples of how to use the functions.

Check Mercury Interactive's Customer Support Web site for updates to the online *Function Reference*.

LoadRunner Tuning Module **Context Sensitive Help** provides immediate answers to questions that arise as you work with LoadRunner Tuning Module. It describes dialog boxes, and shows you how to perform LoadRunner Tuning Module tasks. To activate this help, click in a window and press F1. Check Mercury Interactive's Customer Support Web site for updates to LoadRunner Tuning Module help files.

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

Support Information presents the locations of Mercury Interactive's Customer Support Web site and home page, the e-mail address for sending information requests, and a list of Mercury Interactive's offices around the world.

Mercury Interactive on the Web uses your default Web browser to open Mercury Interactive's home page (http://www.mercuryinteractive.com). This site enables you to browse the knowledge base and add your own articles, post to and search user discussion forums, submit support requests, download patches and updated documentation, and more.

LoadRunner Tuning Module Documentation Set

LoadRunner Tuning Module is supplied with a set of documentation that describes how to:

- install LoadRunner Tuning Module
- use the LoadRunner Tuning Module Console

Using the LoadRunner Tuning Module Documentation Set

The LoadRunner Tuning Module documentation set consists of one installation guide,and one Console user's guide.

In addition to the LoadRunner Tuning Module documentation set, refer to the *LoadRunner Creating Vuser Scripts Guide* and to the *LoadRunner Analysis User's Guide* which were supplied with the LoadRunner documentation set, as described below.

Installation Guide

For instructions on installing LoadRunner Tuning Module, refer to the *LoadRunner Tuning Module Installation Guide*. The installation guide explains how to install:

- the LoadRunner Tuning Module Console—on a Windows-based machine
- Virtual User components—for Windows and UNIX platforms

Console User's Guide

The LoadRunner Tuning Module documentation pack includes one Console user's guide:

The *LoadRunner Tuning Module Console User's Guide* describes how to create and run LoadRunner Tuning Module sessions using the LoadRunner Tuning Module Console in a Windows environment. The Console user's guide presents an overview of the LoadRunner Tuning Module testing process.

Analysis User's Guide

The LoadRunner documentation pack includes one Analysis user's guide:

The *LoadRunner Analysis User's Guide* describes how to use the LoadRunner Analysis graphs and reports after running a session in order to analyze system performance.

Guides for Creating Vuser Scripts

The LoadRunner documentation pack includes one guide for creating scripts.

- The *Creating Vuser Scripts Guide* describes how to create all types of Vuser scripts. When necessary, supplement this document with the online *LoadRunner Function Reference* and the following guide.
- The *WinRunner User's Guide* describes in detail how to use WinRunner to create GUI Vuser scripts. The resulting Vuser scripts run on Windows platforms. The *TSL Online Reference* should be used in conjunction with this document.

For information on	Look here...
Installing LoadRunner Tuning Module	*LoadRunner Tuning Module Installation Guide*
The LoadRunner Tuning Module testing process	*LoadRunner Tuning Module Console User's Guide*
Creating Vuser scripts	*Creating Vuser Scripts Guide*
Creating and running sessions	*LoadRunner Tuning Module Console User's Guide*
Analyzing test results	*LoadRunner Analysis User's Guide*

Documentation Updates

Mercury Interactive is continuously updating its product documentation with new information. You can download the latest version of this document from Mercury Interactive's Customer Support Web site (http://support.mercuryinteractive.com).

To download updated documentation:

1. In the Customer Support Web site, click the **Documentation** link.
2. Select the product name.

   Note that if LoadRunner Tuning Module does not appear in the list, you must add it to your customer profile. Click "My Account" to update your profile.

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

4 Click a document link to download the documentation.

Using the VuGen, Function Reference, and Analysis Documentation

The Virtual User Generator, Function Reference, and Analysis functionality provided with the LoadRunner Tuning Module, are very similar to those provided with the standard LoadRunner.

The documentation for these contains several LoadRunner-specific terms. The following table lists the LoadRunner terms and their parallel ones in the LoadRunner Tuning Module.

LoadRunner	LoadRunner Tuning Module
Scenario	Session Step
Controller	Console
Load Testing	Tuning
LoadRunner Controller User's Guide	LoadRunner Tuning Module Console User's Guide

The Analysis provided with the LoadRunner Tuning Module contains additional graphs that are not available in the standard LoadRunner Analysis. The graphs that are available in LoadRunner Tuning Module, but not in LoadRunner are:

- PeopleSoft Ping Graph
- Distributed Denial of Service Graph
- F5 BIG-IP Graph

The documentation for these graphs is included in the *LoadRunner Tuning Module Console User's Guide*. See Appendix A, "Analysis Graphs" for analysis of these additional graphs.

Typographical Conventions

This book uses the following typographical conventions:

1, 2, 3	Bold numbers indicate steps in a procedure.
➤	Bullets indicate options and features.
>	The greater than sign separates menu levels (for example, **File** > **Open**).
Stone Sans	The **Stone Sans** font indicates names of interface elements on which you perform actions (for example, "Click the **Run** button.").
Bold	**Bold** text indicates method or function names
Italics	*Italic* text indicates method or function arguments, file names or paths, and book titles.
Arial	The Arial font is used for examples and text that is to be typed literally.
<>	Angle brackets enclose a part of a file path or URL address that may vary from user to user (for example, *<Product installation folder>\bin*).
[ ]	Square brackets enclose optional arguments.
{ }	Curly brackets indicate that one of the enclosed values must be assigned to the current argument.
...	In a line of syntax, an ellipsis indicates that more items of the same format may be included.

Part I

Understanding the Tuning Module

1

Introducing LoadRunner Tuning Module

LoadRunner Tuning Module offers a proactive solution for validating and optimizing the capacity of an application and its underlying infrastructure to process business transactions. It combines tuning processes, state-of-the-art technologies, and integrated tuning expertise into a flexible and easy-to-use software package for tuning deployment and production systems.

Mercury Interactive's Safe Deployment System™ (SDS) is key to delivering the advanced capabilities of LoadRunner Tuning Module. SDS uses a combination of technology and a knowledge base that has evolved over years of experience in successful customer deployments. The SDS methodology provides a comprehensive approach to deployment tuning, which includes:

- **Systematic Identification:** LoadRunner Tuning Module automates the process of infrastructure and application optimization by examining the system as a whole and using a logical step-by-step process to identify problems.
- **Problem Isolation:** LoadRunner Tuning Module pinpoints the precise areas where bottlenecks may occur, by using component tests custom-designed by experts.
- **Expert Recommendations:** LoadRunner Tuning Module recommends the appropriate corrective action using a built-in knowledge base and remote tuning capabilities that reflect the learning gathered through years of customer engagements and working with all of Mercury Interactive's key infrastructure vendors.
- **Automated Improvements and Validation:** LoadRunner Tuning Module automates the process of making configuration changes directly to system devices and automatically validates these changes to ensure they have improved the system's performance.

This section describes:

➤ LoadRunner Tuning Module's Features

LoadRunner Tuning Module's Features

LoadRunner Tuning Module's unique features for tuning deployment and production systems include:

➤ System Topology Mapping

➤ Automatic Assignment of Monitors

➤ Configurable Alerts

➤ Canned Scripts

➤ Network Health Check

➤ Log Scanning

➤ Recommended Values Scanning

➤ Tuners

➤ Cartridges

➤ Progress Tracking

➤ Component Value Comparison

➤ Performance Tuning Network

➤ Flexibility and Scalability

System Topology Mapping

LoadRunner Tuning Module's System Topology window lets you draw a map of your system, defining the individual components and their properties. The System Topology window provides the system-wide perspective necessary for tuning multiple-tier systems, as well as the capability to perform drill-down tuning exercises on any component. See "Creating a Topology" on page 27.

Automatic Assignment of Monitors

Complex systems can contain hundreds of counters and statistics. As a result, a user can find it difficult to know which of these are critical for performance. To eliminate this confusion, you can use LoadRunner Tuning Module to automatically select the monitors most critical for each component's performance. See "Accessing the SiteScope Administration Console" on page 46.

Configurable Alerts

Configurable alerts allow for fail-safe control of your tuning exercise, by establishing safe performance thresholds that ensure that your deployment or production system is protected throughout the exercise. If a danger point is reached during a test, LoadRunner Tuning Module can halt or scale down the test, or stop increasing the load on the component.

In addition, LoadRunner Tuning Module's alerts inform you when errors occur, helping you identify bottlenecks on complex multiple-tier systems. Instead of running four or five different tools, LoadRunner Tuning Module locates and displays the cause of the problem, giving you one clear, comprehensive picture. See "Defining Alerts" on page 139.

Canned Scripts

To save you time and effort, LoadRunner Tuning Module includes a large number of pre-configured tuning exercises that are automatically configured for your topology. These are known as *canned scripts*.

LoadRunner Tuning Module's canned scripts save you time, and provide structure and expert guidance on how to begin your tuning sessions. To expand the functionality of LoadRunner Tuning Module's recommended scripts, you can also create custom scripts for your application.

See "Using Built-In Knowledge" on page 83.

Network Health Check

The Network Health Check measures the latency and number of hops between elements in your network. This gives you an indication of the network's overall efficiency. See "Network Health Check Step" on page 226.

Log Scanning

The log scanning feature locates error messages in various system logs, helping you to pinpoint the services and applications that require tuning. See "Scan Logs Step" on page 227.

Recommended Values Scanning

LoadRunner Tuning Module compares the current values on a service or server with the recommended values in LoadRunner Tuning Module's knowledge base, and shows you those values that differ from the recommended ones. See "Scan Recommended Values Step" on page 234.

Tuners

Once bottlenecks have been identified, *tuners* display your components' current settings, recommend which settings to change, and let you tune your components remotely from one location.

Tuners also help you manage your configurations, allowing you to roll back any changes made in LoadRunner Tuning Module, so you can restore your system to its original state. See "Tuning Your System from the Console" on page 655.

Cartridges

LoadRunner Tuning Module's cartridges provide pre-packaged tuning methodologies for specific application environments (for example, Siebel, PeopleSoft, or J2EE environments). They contain repeatable methodologies and extensive knowledge bases geared specifically to your needs. Cartridges shorten the preparation time needed for tuning your application environment. See "Using Built-In Knowledge" on page 83.

Progress Tracking

The Progress Dashboard shows you how your system's performance improves as you monitor and tune it. You define performance goals and track how close you come to achieving them. See "Viewing Step Run Progress" on page 221.

Component Value Comparison

You can compare the values of properties on different hosts and services, and see which properties are different. The comparison feature highlights differences in the configurations of two or more servers, and provides an easy interface for system administrators to synchronize configuration between a laboratory environment to a production environment when migrating. It compares services to ensure see whether they are configured with the same properties or have differences that are not allowable. In load-balanced environments that include multiple servers, it is important to ensure that your web servers, application servers, database servers, and other components are all configured with the same properties when running on similar hardware. Proper and consistent configuration for similar services and hosts results in better load-balancing and system performance.

See "Comparing Hosts and Services" on page 675.

Performance Tuning Network

Mercury Interactive's Performance Tuning Network is a knowledge base relating specifically to performance tuning. The knowledge base gives you tips on locating bottlenecks and tuning your components, and is accessed via LoadRunner Tuning Module's Help menu.

Flexibility and Scalability

To provide maximum flexibility and scalability, LoadRunner Tuning Module includes the entire suite of Mercury Interactive's monitoring technologies. These include multi-layer tuning, capacity planning, security validation under load, and capacity limit and reliability testing.

Part II

Designing a Tuning Session

2

Tuning Session Overview

LoadRunner Tuning Module guides you through the process of tuning your system. This chapter

Typical Workflow

To tune your system, you typically do the following:

1 Identify the tuning goals

These include:

- ➤ desired number of concurrent users
- ➤ expected transaction rate
- ➤ expected response time
- ➤ maximum acceptable error rate

2 Create or open a session

Create a new session from a cartridge (a template for a session), or open an existing session. See Chapter 3, “Getting Started With Sessions.”

3 Define the topology

Use LoadRunner Tuning Module’s topology mapping tool to define the production system’s components and the connections between them. See Chapter 5, “Creating a Topology.”

4 Assign monitors

Assign monitors to track performance measurements on various topology elements. See Chapter 25, “Online Monitoring.”

5 Define alerts

Define alerts to measure the state of your systems and establish safe performance thresholds. See Chapter 13, "Defining Alerts."

6 Create session steps

Although the session that you create from a cartridge includes session steps of various types, you can create additional steps to suit your purposes. These can be any of the available step types: Execution, Manual, Scan Logs, Scan Recommended Values, or Network Health Check. See Chapter 7, "Adding Session Steps."

7 Perform manual step actions

Perform the actions defined in the manual steps. These are typically instructions for setting up your environment. See Chapter 8, "Creating and Using Manual Steps."

8 Configure Execution steps

Configure the Execution steps (those you created and those that were provided in the cartridge) to test the components suspected of causing bottlenecks. This includes:

- choosing scripts to run tests against components
- choosing profiles for each session step

See Chapter 9, "Creating and Configuring Execution Steps."

9 Execute a Benchmark execution step

Test the components of your system and establish baseline performance measurements before tuning your system. LoadRunner Tuning Module's monitors and analysis display your performance in real time. During the session execution, LoadRunner Tuning Module records your application's performance under different loads.

10 Perform additional tuning functions

Perform one or more of the following functions:

- Check Network Health—Perform a network health check to measure the latency and number of hops between connected components. See "Network Health Check Step" on page 226.

- Scan Log Files—Use a Scan Log Files step to locate error messages in the various log files. This can indicate problematic components. See "Scan Logs Step" on page 227.
- Scan Recommended Values—Use a Scan Recommended Values step to compare the current values on a machine or service with the recommended values in LoadRunner Tuning Module's knowledge base. See "Scan Recommended Values Step" on page 234.

Note: You may want to perform one or more of these functions at the beginning of your tuning session, before executing any other steps, to quickly obtain a picture of your system's performance and the way its components are configured.

---

11 Execute additional Execution steps

Run other Execution steps to test various components and services.

12 Analyze performance

View offline analysis data collected while you ran the session steps. Use this data to analyze application performance, and identify problematic components and services.

13 Tune the problematic components

Tune the problematic components based on the Execution step results. You can use LoadRunner Tuning Module's integrated tuners to make changes to system component properties from the Console machine. See Chapter 45, "Tuning Your System from the Console."

14 Run the Benchmark step again

Run the benchmark step again to validate the improvement in performance.

15 Repeat the tuning cycle

Continue to run session steps and tune the system based on the session step results, until you achieve your performance goal.

3

Getting Started With Sessions

This chapter describes how to invoke LoadRunner Tuning Module and open or create a session.

This chapter discusses:

- Creating or Opening a Session

About Getting Started

A LoadRunner Tuning Module session consists of a number of session steps of different types, and is based on a cartridge. A cartridge contains a group of steps, along with suggested settings for monitors and alerts, and recommended values for your specific environment. It is based on LoadRunner Tuning Module's knowledge base. For more information about cartridges, see "Using Built-In Knowledge" on page 83.

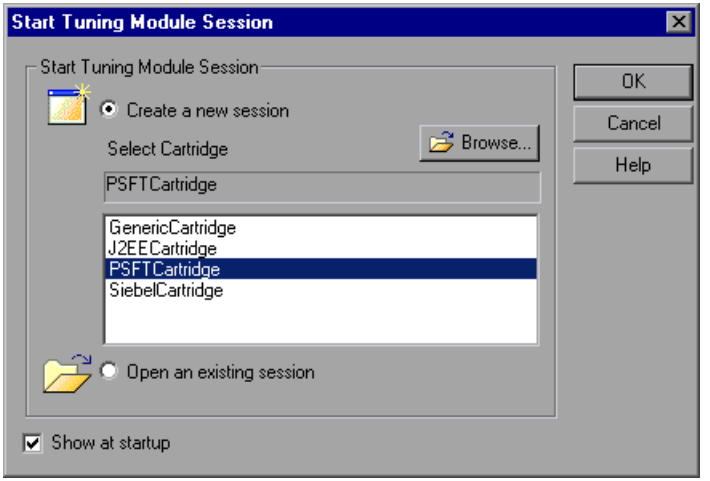
After choosing the cartridge, you then customize it to your specific environment.

Creating or Opening a Session

You first invoke LoadRunner Tuning Module, and then specify whether you want to create a new session or open an existing one.

To create or open a LoadRunner Tuning Module session:

1 Choose **Start** > **Programs** > **LoadRunner Tuning Module** > **Console**. LoadRunner Tuning Module displays the Start Session dialog box:



2 Choose whether to create a new session or open an existing session.

3 Click **OK**. The Session tab is displayed, showing you the steps that the session contains.

See "Viewing the Session Tab," on page 19.

Understanding the Start Tuning Module Session Dialog Box

The Start Tuning Module Session dialog box allows you to create a new session immediately when invoking LoadRunner Tuning Module, or to open an existing session.

Create a new session: Click this radio button and choose a cartridge from the list. If the cartridge does not appear in the list, click **Browse** to locate it. When you click **OK**, LoadRunner Tuning Module creates a new session based on the cartridge. For information about cartridges, see "Using Built-In Knowledge" on page 83.

Open an existing session: Click this radio button and click **OK** to display the Open Session dialog box. Locate the session and click **Open**.

The **Show at startup** box allows you to specify whether LoadRunner Tuning Module should display this window whenever the Console is invoked.

4

Viewing the Session Tab

This chapter describes what you see in the Session tab when a session has been opened or created.

This chapter discusses:

- Viewing Session Information
- Viewing Step Information
- Status Bar
- Step Actions

About the Session Tab

When the Console first displays the Session tab, the upper left pane contains the Session Step window. This is a tree view of the steps that have been defined for the session.

The root element of the tree represents the session, which is based on a cartridge. The child elements are the session steps and (for Execution steps) the scripts that they contain.

The following information is always displayed, regardless of whether you are viewing information about the session or about a step:

- **Description:** Session or step description. If the session is selected, this explains what the session contains and how to use the various steps. If a step is selected, the text gives information about the step and guidelines for its use. You can edit the text in this field. When you save a cartridge or session, the modified text replaces the original text.
- **Topology:** A diagram of your system topology, if a topology has been defined. When you double-click inside this window, the System Topology window opens. See "Creating a Topology" on page 27 for details.

Viewing Session Information

To view information about the session:

1. Click the tree's root element to select the session element. The Console displays the following additional information:

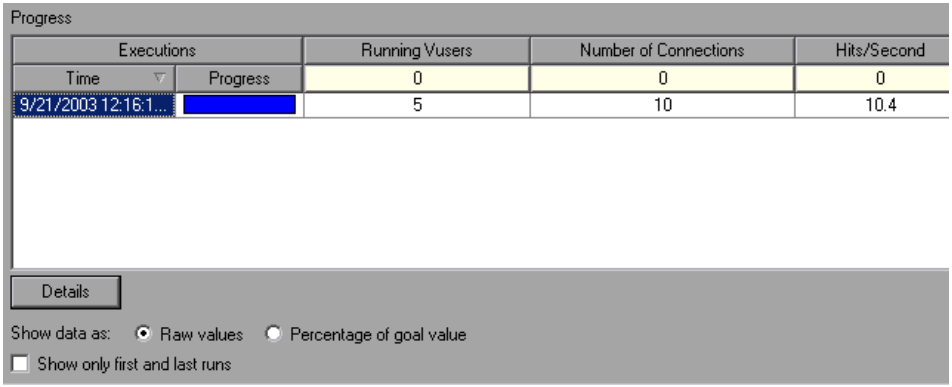
- **Session Goals:** The default goals for the session are displayed, as in the following example:

Session Goals

Goal	Goal Condition	Goal Value
Running Vusers	>=	5.00
Average Transaction Response Time	<=	0.50
Number of Connections	>=	10.00
Hits/Second	>=	20.00
Connections/Sec	>=	10.00

You can change the goal conditions and values as necessary.

➤ **Progress:** This section of the window displays the Progress Dashboard, which shows you to what extent the goals have been achieved during the session.



See "Viewing Step Run Progress" on page 221 for details.

Viewing Step Information

To view information about a specific step:

Click the step name element in the tree to select it. In addition to the Description and Topology fields, your view of the right side of the Session tab depends on the type of step currently selected

For information on this type of step...	See this section
Manual	"Creating and Using Manual Steps" on page 51.
Execution	"Creating and Configuring Execution Steps" on page 55.
Network Health Check	"Advanced Tuning Functions" on page 225.
Scan Logs	"Scan Logs Step" on page 227.
Scan Recommended Values	"Scan Recommended Values Step" on page 234.

If you select an Execution step, expanding the step name displays its Vuser scripts.

Status Bar

The status bar displays the following information:

- **Errors**. Clicking the number of errors displays the Output window, which shows you a list of the error messages generated during the session step's execution.
- **Alerts**. Clicking the number of alerts displays the Output window, which shows you a list of alerts triggered during the session step's execution.
- **Step status**
- **Active Vusers**
- **Elapsed time**
- **Progress**. Clicking the progress bar displays the Progress Dashboard. See "Viewing Step Run Progress" on page 221.
- **Failed and Passed Transactions**. When you click the number of transactions, the Transactions window displays statistics on the transactions that have taken place during the step's execution. The status of the selected transaction (passed or failed) is highlighted in the Transactions window. When you click the number of errors or alerts, the appropriate Output window is displayed.

Step Actions

In the Session tab, you can perform the following actions that affect session steps (note that some functions are only available in certain types of steps):

 **Add Step**	Adds a step to the session. See “Adding Session Steps” on page 48.
 **Remove Step**	Removes a step from the session. See “Managing Steps” on page 49.
 **Move to Next Step**	Advances to the next step in the session.
 **Edit Schedule Settings**	Schedules the session step by defining a manual or goal-oriented profile. See “Scheduling Session Steps” on page 119.
 **Execute Session Step**	Runs the selected Execution step. See “Running an Execution Step” on page 189.
 **Configure**	Opens the Step Configuration dialog box. See “Understanding the Step Configuration Dialog Box” on page 67.
 **Execute Benchmark Step**	Executes the step that you specified in the Add Step dialog box, or by choosing Set As Benchmark when right-clicking a step in the Session tree. See “Understanding the Add Step Dialog Box” on page 61.
 **Stop Gradually**	Gradually stops the Execution step. See “Running an Execution Step” on page 191.
 **Stop Immediately**	Stops the Execution step immediately. See “Running an Execution Step” on page 191.

 **Add Vusers**	Opens the Run/Stop Vusers dialog box where you can add Vusers to a running step. See "Understanding the Run/Stop Vusers Dialog Box" on page 196.
 **View and Control Vusers**	Opens the Vuser Management window. See "Understanding the Vuser Management Window" on page 199.
 **Add/Remove Graphs**	Opens the Select Online Graphs dialog box. See "Understanding the Select Online Graphs Dialog Box" on page 298.
 **Analyze Results**	Invokes Analysis to analyze the session step results.
 **Insert New Task**	Allows you to add a task to a manual step. See "Creating a Manual Step" on page 52.
 **Delete Task**	Allows you to remove a task from a manual step. See "Creating a Manual Step" on page 52.
 **Move Task Up**	Moves a task up in the table of a manual step. See "Creating a Manual Step" on page 52.
 **Move Task Down**	Moves a task down in the table of a manual step. See "Creating a Manual Step" on page 52.
 **Select All Tasks**	Selects all the tasks in the table of a manual step. See "Creating a Manual Step" on page 52.
 **Check Network Health**	Executes a Network Health Check step. See "Network Health Check Step" on page 226.
 **Scan Recommended Values**	Executes a Scan Recommended Values step. See "Scan Recommended Values Step" on page 234.

 **Save to File**	After executing a Scan Recommended Values step, saves a report of the properties whose values differ from the recommendations. See “Saving Recommended Values Reports” on page 237.
 **Start Log Scan**	Starts scanning log files for errors. See “Scanning Log Files” on page 228.
 **Stop Log Scan**	Stops scanning log files. See “Scanning Log Files” on page 228.
 **Scan Rules**	Opens the Scan Rules dialog box, enabling you to specify how the Console should search for errors when scanning log files. See “Scanning Log Files” on page 228.
 **View Filters**	Opens the View Filters dialog box, enabling you to specify the way the log data is displayed in a Scan Logs step. See “Scanning Log Files” on page 228.

5

Creating a Topology

Before you can test and measure your system's performance, you need to create a topology diagram and map its components to the elements of your system that are used in the business process. This chapter describes how to create a topology.

This chapter discusses:

- Building a Topology Diagram
- Setting the Component Properties

About Creating a Topology

The topology is a graphical representation of the system architecture.

Defining the topology consists of specifying the components and their types (for example, whether a component is a database, Web, or application server). You also define the connections between the components.

The topology also includes other hardware, such as routers, firewalls, and load generator machines.

LoadRunner Tuning Module provides a variety of topology templates for typical network configurations. Each template contains a number of different components.

For a list of LoadRunner Tuning Module's topology templates, see "Understanding the Topology Templates Window" on page 32.

You can use one of these templates as is or customize it. When you add items to the topology, LoadRunner Tuning Module arranges the elements in the most common layout. You can rearrange this layout or delete any unnecessary components.

Building a Topology Diagram

After you have created a session, you proceed to building your topology diagram.

Before building a topology diagram, make sure you have a clear understanding of your system architecture. An incorrect representation of your system will result in inaccurate results.

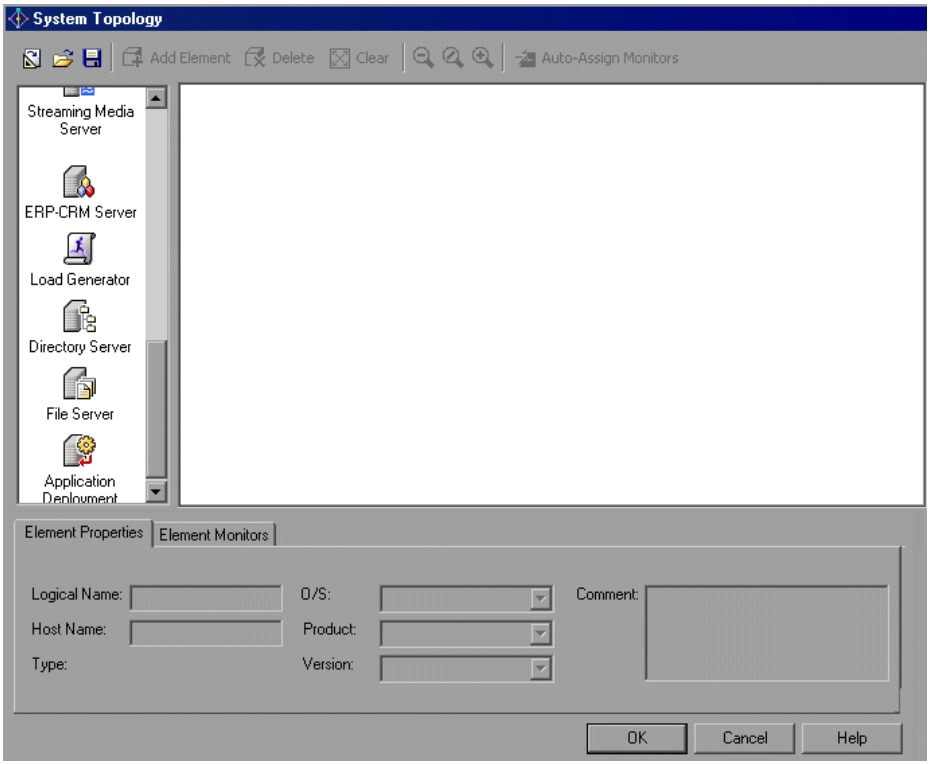
Using LoadRunner Tuning Module's design window, you can drag topology elements from the design palette into the topology diagram workspace.

You can map a physical component to more than one logical topology element. For example, if a computer hosts both a Web server and a database server, you can define two elements—a Web server element and a database server element—and map them to the same computer.

You can use an existing topology, base your topology on a template, or create a new topology from scratch. LoadRunner Tuning Module allows you to modify templates and existing topologies to suit your specific needs.

1 If the Console is running but the System Topology window is not displayed, click the **Topology** button or choose **Tools** > **System Topology** to open it.

The System Topology window is displayed:



Note: You can also open the System Topology window by double-clicking the white space in the **Session** tab's topology pane.

---

2 Select the topology elements that reflect your environment, and assign monitors and alerts as required.

3 Click **OK**. The Session tab is displayed.

Understanding the System Topology Window

The System Topology window allows you to specify your system's components, and assign them monitors and alerts. The System Topology window lets you define the following types of components: Web servers, application servers, mail servers, ftp servers, groupware servers, firewalls, database servers, routers, load generator machines, DNS servers, load balancers and streaming media servers. You create a graphic representation of the topology by dragging the relevant components from the design palette into the topology diagram workspace, and map the graphic elements to your system's components.

Design palette: Contains system components. To add a component, drag it from the design palette to the topology diagram, or use the Add Element button.

Add Element: Adds a selected component from the design palette to the topology diagram workspace.

Template: Opens the Load Topology Template dialog box (see "Understanding the Topology Templates Window" on page 32). You can select one of the provided topology templates.

Open Topology File: Imports an existing topology from a specified file location, and displays it in the topology diagram workspace.

Save As Topology File: Saves the topology diagram for use in another session. Specify a file name and the location where you want to store the diagram.

Clear: Clears all the contents of the topology diagram workspace.

Delete: Deletes the selected component from the topology diagram workspace.

Auto Assign Monitors: Assigns monitors automatically to selected components, based on the component types. Use the <Ctrl> key to select more than one component, or use <CTRL>+A to select all the components. See "Automatically Assigning Monitors and Alerts" on page 37.

Zoom-In: Zooms in on the Topology components.

Zoom-Out: Zooms out from the Topology components.

Zoom to Original Size: Shows you a view of the topology components without zooming.

Element Properties Tab: Allows you to define the server properties for each component in the topology diagram. For more information, see "Understanding the Element Properties Tab" on page 35.

Element Monitors Tab: Allows you to select the measurements that you want to monitor for each component. For more information, see "Understanding the Element Monitors Tab" on page 45.

To import an existing topology diagram:

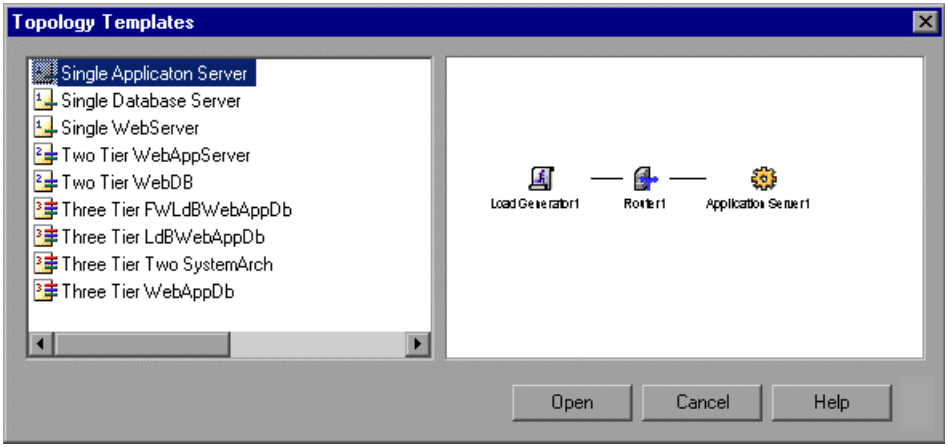
1 Click **Open Topology File** and browse to the desired topology (a file with a *.tpl* extension).

2 Click **Open**. LoadRunner Tuning Module displays your topology.

Creating Topology Diagrams from Templates

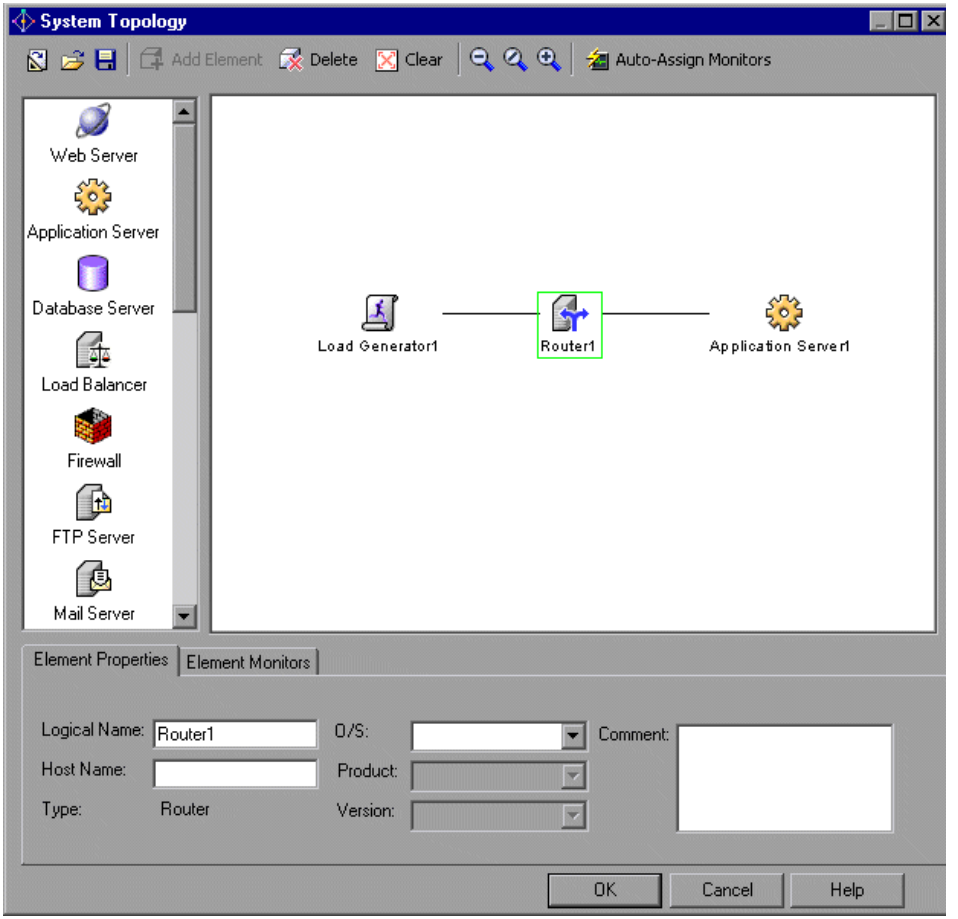
To create a topology diagram from a template:

1 Click **Topology Templates** to open the Topology Templates dialog box.



2 Select a template and click **Open**.

The topology diagram you selected appears in the right pane (the topology diagram workspace), and the topology elements appear in the left pane, as in the following example:



3 Customize the diagram as needed.

Understanding the Topology Templates Window

The Topology Template window enables you to create a topology diagram from one of the templates provided with LoadRunner Tuning Module.

Following is a list of LoadRunner Tuning Module's topology templates:

- Single Application Server
- Single Database Server
- Single Web Server
- Two Tier WebAppServer
- Two Tier WebDB
- Three Tier FWLdBWebAppDb
- Three Tier LdBWebAppDb
- Three Tier Two System Arch
- Three Tier WebAppDb
- Siebel Two Tier Simple
- Siebel Two Tier Multi Server
- Siebel Two Tier Multi Server Citrix
- Siebel 7.x Simple
- Siebel 7.x Multi Server
- Siebel 7.x Multi Server Citrix
- PeopleSoft Two Tier Simple
- PeopleSoft Two Tier Complex
- PeopleSoft 8.x Simple
- PeopleSoft 8.x Multi Server
- PeopleSoft 8.x Multi Server Citrix
- PeopleSoft 8.x Multi Server LB Secured

To create or modify a topology:

1. To add a component to a topology, select the component in the left pane and click **Add Element**, or drag the component from the left pane into the topology diagram workspace.
2. To change a component's position, click the component and drag it to its new position within the topology.

3 To connect components in the topology, drag a line from one component to the other.

4 To delete a component, select the component by clicking it in the diagram, and click the **Delete** button or press your keyboard's Delete key.

5 To delete all the components from the topology diagram, click **Clear**.

To zoom in on the System Topology window:

➤ Click **Zoom In**, and use the directional scroll bars on the side of the System Topology window to focus on the relevant components.

To zoom out:

➤ Click **Zoom Out,** and use the directional scroll bars on the side of the System Topology window to focus on the relevant components.

To restore the original size of components in the System Topology window:

➤ Click **Zoom to Original Size**.

To save the topology diagram for use in another session:

1 Click **Save As Topology File**.

2 Browse to the desired location, specify a filename with a *.tpl* extension, and click **Save**.

Setting the Component Properties

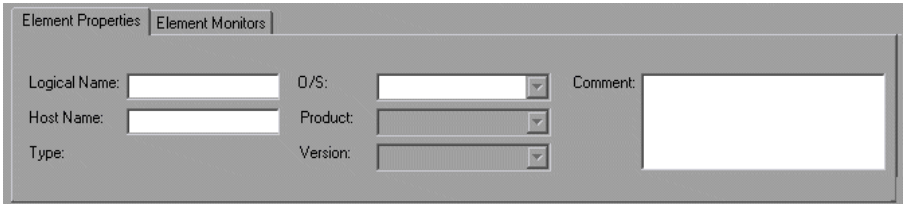
After you create a topology diagram, you specify the properties for each component. LoadRunner Tuning Module builds a tuning session based on these settings.

You can specify the following properties for each component:

- logical name
- host name
- operating system
- product
- version

To specify a component's properties:

1. Select the component whose properties you want to specify.
2. Click the **Element Properties** tab.



3. Specify the component's properties in the appropriate fields.

Understanding the Element Properties Tab

You use the Element Properties tab to define the server properties for each component in the topology diagram.

Type: Displays the type of component selected in the topology diagram workspace.

Logical Name (Optional): The name that you want LoadRunner Tuning Module to use for the component when it appears in the topology diagram, graphs, reports and other places. You can use up to 255 characters. You can

use the logical name to indicate the task performed by the component, for example, "Backup_DB_Server."

Host Name: The component's host name, up to 50 characters.

Notes:

- If you specify the component's logical name but not its host name, LoadRunner Tuning Module copies the logical name into the Host Name field. If you specify neither host name nor logical name, LoadRunner Tuning Module will not allow you to select monitors for the element.
- If you delete a component's host name and then change the focus to another element or exit the topology window, LoadRunner Tuning Module warns you that you are about to remove all of the monitors and alerts previously assigned to the component.

O/S: Select your component's operating system from the list.

Product: Select the product or vendor for the selected component from the list.

Version: Select the application version from the list.

Comment (Optional): Add any useful information about the component, up to 255 characters.

6

Selecting Monitors in the Topology Window

After defining your components' properties in the Topology window, you can select the monitors and alerts used to track their performance.

- Automatically Assigning Monitors and Alerts
- Selecting Monitors Manually
- Accessing the SiteScope Administration Console

About Selecting Monitors

Selecting monitors includes specifying the measurements that you want to monitor on each component. You can use the Auto-Assign feature to automatically assign monitors and alerts, or select and assign them manually.

Automatically Assigning Monitors and Alerts

LoadRunner Tuning Module's Auto Assign feature assigns monitors automatically to selected components, based on each component's type. This allows you to skip the task of choosing monitors for your components.

Note: The Auto Assign feature is currently enabled only if the Console machine is running an English version of Windows.

For example, if your component is an Apache Web Server, the Auto Assign feature assigns to your component all the monitors for all the Apache measurements.

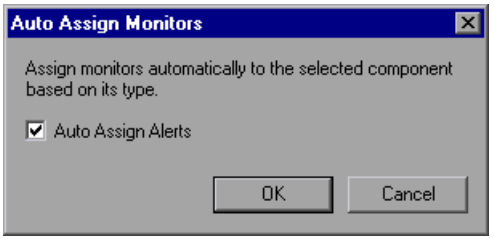
Tip: For details of LoadRunner Tuning Module's monitors, see "Monitoring a Session" on page 287.

In addition to automatically assigning monitors, you can also automatically assign alerts. This assigns a default set of alerts for various measurements, based on definitions in LoadRunner Tuning Module's knowledge base. For information on alerts, see "Defining Alerts" on page 139.

To automatically assign monitors and alerts to components:

1 In the topology diagram, select the components whose properties you want to monitor. Use the <Ctrl> key to select more than one component, or use <CTRL>+A to select all the components.

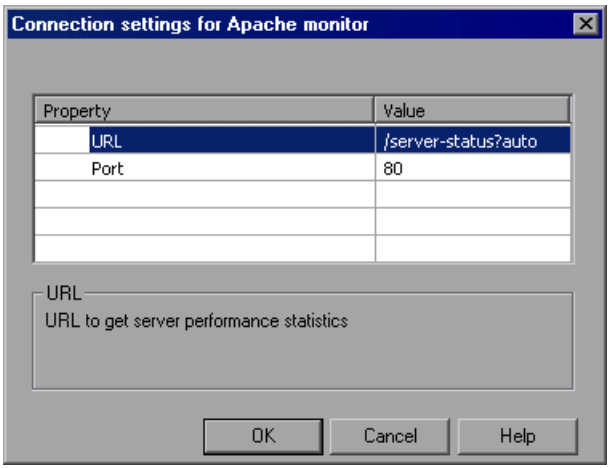
2 Click **Auto Assign Monitors**. Alternatively, you can right-click a component and click **Auto Assign Monitors**. LoadRunner Tuning Module displays the Auto Assign Monitors dialog box:



3 To automatically assign alerts, select **Auto Assign Alerts**.

4 Click **OK**. If the monitors that are being assigned have configurable parameters or require insertion of parameters, LoadRunner Tuning Module displays a Connection Settings dialog box for each of these monitors. The dialog box displays parameters that are specific to the monitor.

It typically allows you to insert values and change one or more of the displayed default values, as in the following example:



For more information on the Connection Settings dialog box, see "Understanding the Connection Settings Dialog Box." below.

5 After making your changes, click **OK**.

LoadRunner Tuning Module assigns the monitors to the selected components, and displays them in the Element Monitors section of the Topology window. A monitor sign is appended to the component's icon in the Topology diagram to indicate that the monitors have been assigned. For example, a Web server that has had monitors assigned to it appears as follows: 

6 If you chose to assign alerts, you can view the assigned alerts by clicking the **Alerts Definition** button.

Understanding the Connection Settings Dialog Box

The Connection Settings dialog box allows you to specify connection settings for a monitor when you choose the Auto Assign feature.

This dialog box appears if a monitor that is being assigned has configurable parameters or requires additional data. The displayed parameters depend on the particular monitor that LoadRunner Tuning Module is assigning to the selected component. You can typically insert values and change one or more of the displayed default values.

The dialog box displays one or more of the following parameters, depending on the monitor:

- Username
- Password
- Server Name
- URL
- Port

Selecting Monitors Manually

After you define each component's properties, you select the measurements that you want to monitor for the component.

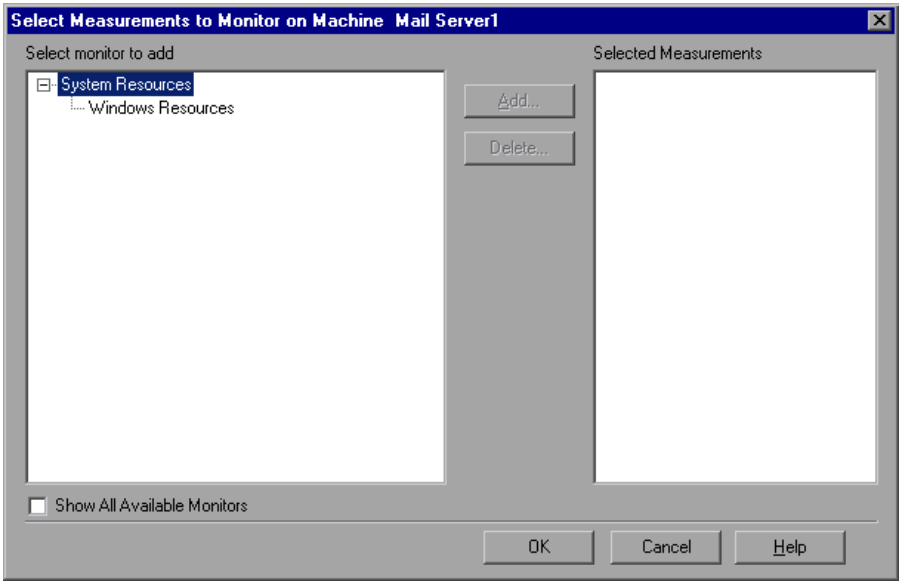
Note: You need to define the component properties before selecting the monitors, since LoadRunner Tuning Module uses these properties to locate the available monitors in its knowledge base.

To select a measurement to monitor:

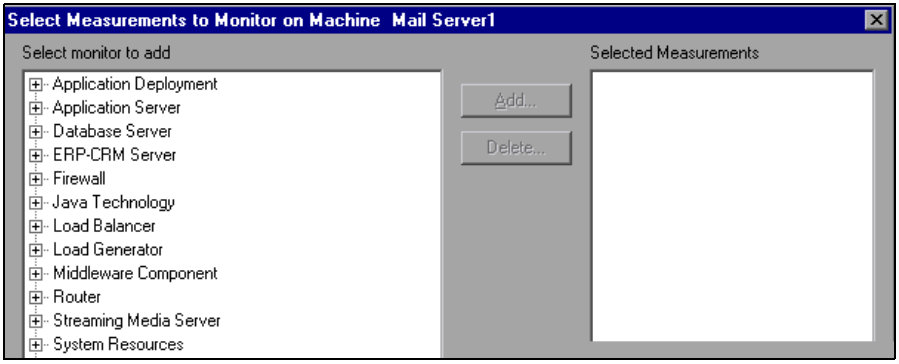
1. In the topology diagram, select the component whose properties you want to monitor.
2. Click the **Element Monitors** tab and then click **Add Monitor**. The **Select measurements to monitor** dialog box opens. For detailed information about this dialog box, see "Understanding the Select Measurements to Monitor Dialog Box" on page 45.

Note: To select monitors and measurements when the System Topology window is not displayed, click the **Monitors** button on the main toolbar. See "Choosing Monitors and Measurements" on page 292.

If the LoadRunner Tuning Module knowledge base contains a list of monitors for your element, LoadRunner Tuning Module displays only those monitors available for the component in its specified operating system, product and version, as in the following example:

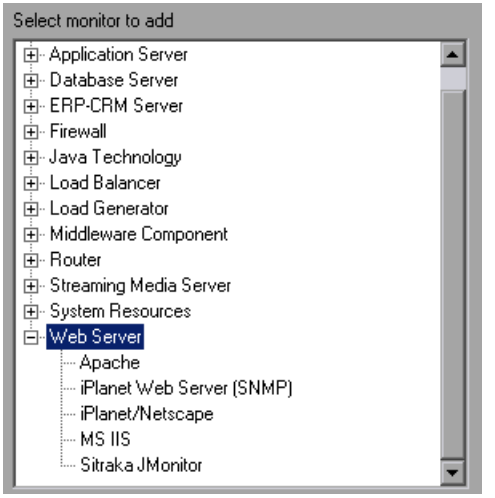


If the knowledge base does not contain a list of monitors for the specified element (possibly because you haven't specified all of the element's properties), LoadRunner Tuning Module displays all the monitors in its knowledge base:



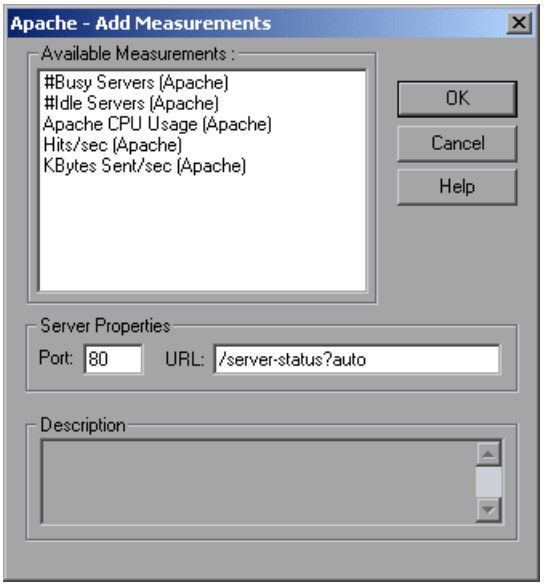
If not all of LoadRunner Tuning Module's monitors are displayed, you can display them by checking the **Show All Available Monitors** option.

The full list of available monitors appears as a list of component types. If the monitor you need is not visible, expand the appropriate component type to display the list of components in your category. For example, clicking Web Server shows you the list of Web Server applications whose measurements you can monitor:



Click a component to select it and then click **Add**, or double-click the component. For most component types, LoadRunner Tuning Module opens a dialog box with a list of the component's available measurements.

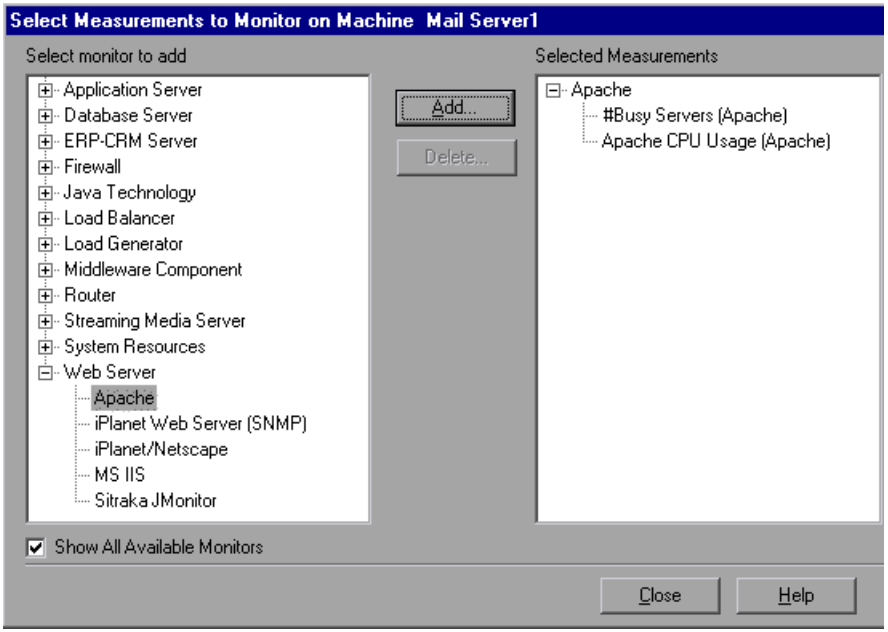
For example, if you choose Apache, LoadRunner Tuning Module displays the following list of Apache measurements that you can monitor:



Tip: For information on how to specify the measurements specific to each component, see the appropriate chapter in the section "Monitoring a Session" on page 287.

---

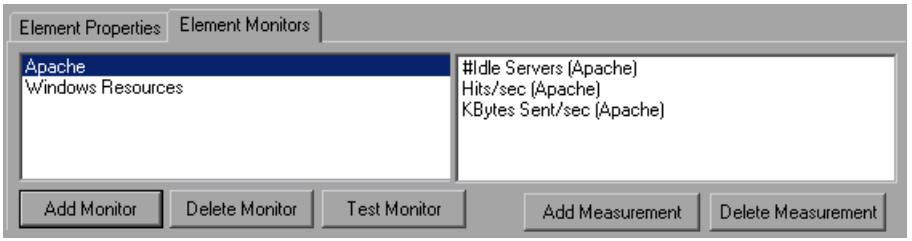
3 Select the measurements that you want to monitor, and click **Add**. LoadRunner Tuning Module adds the measurements to the Selected Measurements pane.



4 To delete a measurement from the Selected Measurements pane, select the measurement and click the **Delete** button.

5 Repeat steps 1 through 3 for all the components that you want to monitor, and then click **Close**.

LoadRunner Tuning Module displays the newly added monitors in the Element Monitors section.



Understanding the Element Monitors Tab

The Element Monitors tab enables you to select the measurements to monitor for your servers. Note that you must define the server properties before selecting the monitors, since LoadRunner Tuning Module connects to the servers to determine which monitors are available.

The left pane displays element monitors. If you select an element monitor, the measurements it will monitor are displayed in the right pane.

Add Monitor: Opens the Select Measurements to Monitor dialog box.

Delete Monitor: Deletes the selected element monitor from the monitors window.

Test Monitor: This button appears for some monitors. Tests whether you can access the monitor.

Add Measurement: Opens the Add Measurements dialog box for the selected monitor.

Delete Measurement: Deletes the selected measurement from the list of measurements in the right pane.

Understanding the Select Measurements to Monitor Dialog Box

The Select Measurements to Monitor dialog box lets you select the monitor type you want to measure. Note that you must define the server properties before selecting the monitors, since LoadRunner Tuning Module connects to the servers to determine which monitors are available.

Select monitor to add: Select the monitor whose measurements you want to view.

Add: Opens a dialog box (for most server types) with a list of the available measurements.

Show All Available Monitors: Check this box to display a list of all of LoadRunner Tuning Module's monitors (not just the ones available for the component in its specified operating system, product and version).

Delete: To delete a measurement from the Selected Measurements pane, select the measurement and click the **Delete** button.

Accessing the SiteScope Administration Console

If you define a SiteScope Server element, LoadRunner Tuning Module allows you to access the SiteScope Administration Console via the Topology window.

To access the SiteScope Administration Console:

➤ Right-click the SiteScope Server element in the Topology window, and choose Open SiteScope. The SiteScope Administration Console opens in your browser, as in the following example.

For information about SiteScope, refer to the SiteScope documentation.

7

Adding Session Steps

The new session that you created from a cartridge includes a series of steps. After you define a topology, you can add steps to the session and specify the order in which they should be executed. This chapter describes how to create and manage session steps.

This chapter discusses:

- Step Types
- Adding Session Steps
- Managing Steps

About Adding Session Steps

Although sessions you create from cartridges contain predefined steps, you frequently need to add steps to the sessions. The following sections explain how to add, manage, and configure these steps.

Step Types

A session can include different types of steps:

- **Execution steps:** These execute a script or scripts against a specified server, to emulate a business process.
- **Manual steps:** These provide a check list of tasks for you to perform.
- **Network Health Check steps:** These measure the latency and number of hops between elements in your network.

- **Scan Logs steps:** These check the server logs for error messages.
- **Scan Recommended Values steps:** These compare the parameter values on the servers with the recommended values.

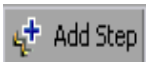
When you create a step, you specify its type.

Adding Session Steps

Before creating Execution session steps, you need to create or load a topology (see Chapter 5, "Creating a Topology"). In addition, it is advisable to create or load a topology before using other types of session steps (see below).

Since any session that you create is based on a cartridge, it already contains session steps. You can add steps of different types to the session.

To add a session step:



1. Click **Add Step** or choose **Session** > **Add Step**, to display a menu list of step types.
2. Choose the type of step that you want to create. The resulting screen depends on the type of step you are creating:
   - **Manual:** Provides a check list of tasks for you to perform independently of LoadRunner Tuning Module. After you perform each task, you check a box to indicate its completion. These steps guide you through tasks that LoadRunner Tuning Module can't perform by itself. When you choose **Session** > **Add Step** > **Manual**, the Console displays a table where you specify the tasks that the step contains. See "Creating and Using Manual Steps" on page 51.
   - **Execution**: Executes scripts against a specified server. When you choose **Session** > **Add Step** > **Execution**, the Console displays the Add Step dialog box, where you choose the script or scripts that the step will execute, and the servers against which the scripts will be run. See "Creating and Configuring Execution Steps" on page 55.
   - **Network Health Check:** Checks the latency of the communication between two machines. It may also show the communication routing details (the hops and routes that the packets travel). When you choose

Session > **Add Step** > **Network Health Check**, the Console displays a window that allows you to measure the latency. See "Advanced Tuning Functions" on page 225.

- **Scan Logs:** Searches log files for specified strings. When you choose **Session** > **Add Step** > **Scan Logs**, the Console displays a window that allows you to choose the machines and services and to execute the scan. See "Scan Logs Step" on page 227.
- **Scan Recommended Values:** Compares the parameters of a machine or service with the recommended values, allowing you to print the results of the comparison.

  When you choose **Session** > **Add Step** > **Scan Recommended Values**, the Console displays a window that allows you to execute the scan. See "Scan Recommended Values Step" on page 234.
- **Import LoadRunner Scenario:** Lets you create an execution step from an existing LoadRunner scenario. See "Importing LoadRunner Scenarios" on page 64.

Managing Steps

After you add steps to your session, you can manage them in several ways. The step commands are available from the right-click menu and the toolbar.

Remove a step: Select a step, and then click **Remove Step** or choose **Remove Step** from the right-click menu.

Rename a step: Right-click the step and choose **Rename Step** to enable editing of the step name. Specify a new name for the step.

Duplicate a step: Right-click the step and choose **Duplicate Step**. A new step, which is a copy of the original step, is added to the session tree below the original step.

Move a step up: To move a step up in the session tree, right-click it and choose **Step Up**.

Move a step down: To move a step down in the session tree, right-click it and choose **Step Down**.

8

Creating and Using Manual Steps

Manual steps are typically used for listing tasks that you need to perform in order to set up your environment for tuning. When you create a session from a cartridge, the first steps in the session are usually manual steps.

This chapter discusses:

- Creating a Manual Step
- Using Manual Steps

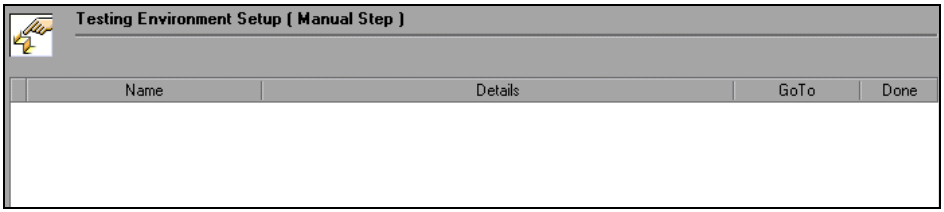
About Defining a Manual Step

A manual step consists of a check list of tasks that you need to perform independently of LoadRunner Tuning Module.

When you create a manual step, the Session tab displays a table for entering the tasks that the step includes.

Creating a Manual Step

When you choose to create a manual step, the Console prompts you to enter the step name. After you enter the name, the Console displays an empty table in the right part of the Session tab.



Each row of the table is used to represent a task that needs to be performed manually.

To add a task to the step:

1 Click **Insert New Task**, or right-click inside the table and choose **New Task**.

The Console adds an empty row to the table.

2 Click the Name field and enter the name of the task.

3 Click the Details field and enter an explanation of what the user needs to do.

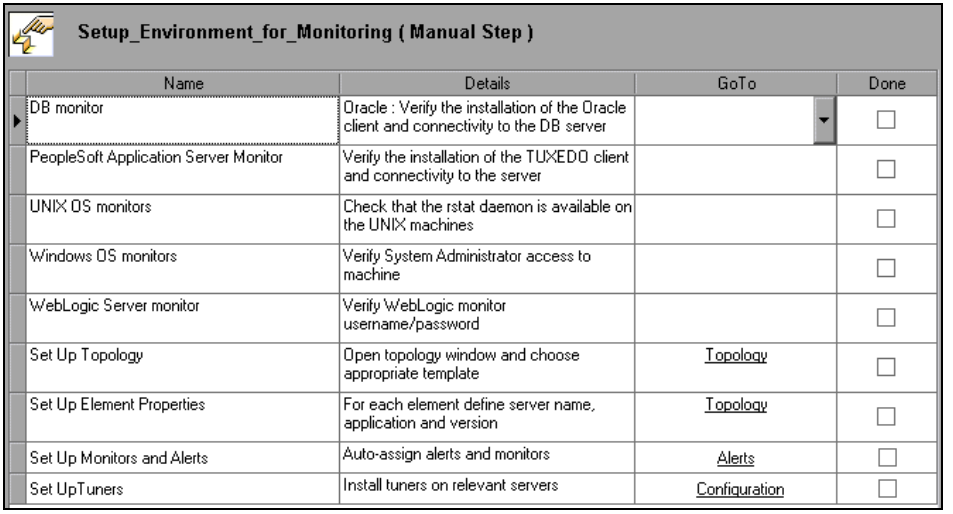
4 Click the arrow in the Goto field to open a drop-down list of destinations. Selecting a destination causes it to be displayed as a link in the GoTo field. When you subsequently click the link, LoadRunner Tuning Module takes you to the specified destination. Following are the available links and their destinations:

Link	Destination
Topology	Topology window. See "Understanding the System Topology Window," on page 30.
Configuration	Configuration tab. See "Understanding the Configuration Tab," on page 656.

Link	Destination
Alerts	Alerts window. See "Understanding the Alerts Window," on page 150.
Monitors	Monitor Configuration dialog box. See "Understanding the Monitors Configuration Dialog Box," on page 294.

5 Select the destination. It now appears in the Goto column.

6 Repeat steps 1 through 5 for all the tasks that you want to add to the step. The table now shows a list of tasks in the step:



To modify tasks in the table:

1 Select the task or tasks by clicking the column to the left of the task name. A right-arrow appears in the column, indicating that the task is selected.

To select all tasks in the table, click **Select All Tasks**.

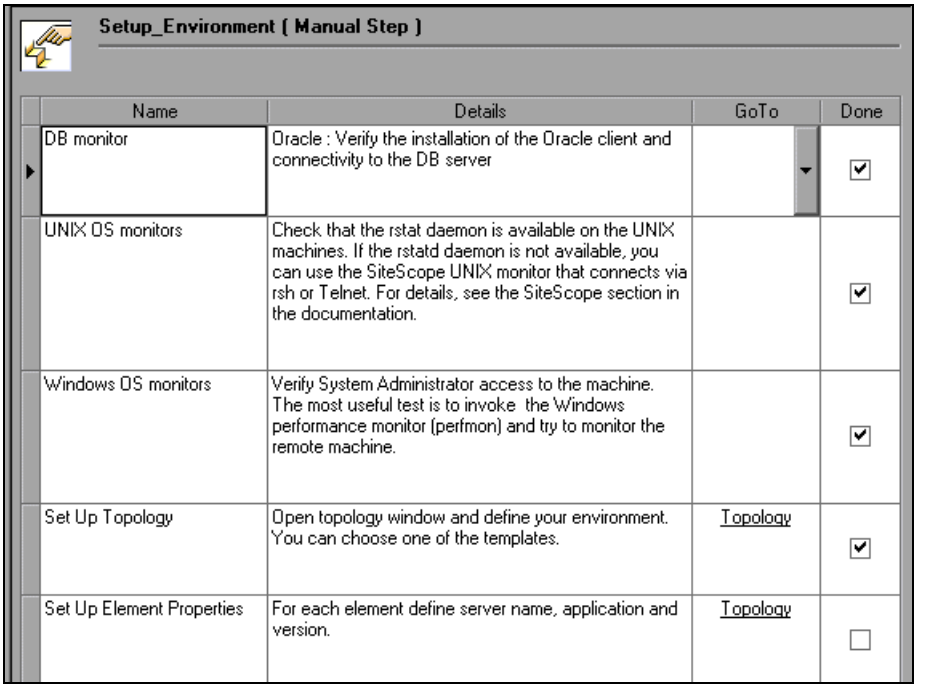
2 To delete, click **Delete Task**. Alternatively, you can right-click the task and choose **Delete Tasks.**

3 To move the task up in the table, click **Move Task Up**, or right-click the task and choose **Move Task Up**.

4 To move the task down in the table, click **Move Task Down**, or right-click the task and choose **Move Task Down**.

Using Manual Steps

Perform each task in the order in which it appears in the table. When you complete a task, check it off in the Done column.



Setup_Environment (Manual Step)

Name	Details	GoTo	Done
DB monitor	Oracle : Verify the installation of the Oracle client and connectivity to the DB server		☑
UNIX OS monitors	Check that the rstat daemon is available on the UNIX machines. If the rstatd daemon is not available, you can use the SiteScope UNIX monitor that connects via rsh or Telnet. For details, see the SiteScope section in the documentation.		☑
Windows OS monitors	Verify System Administrator access to the machine. The most useful test is to invoke the Windows performance monitor (perfmon) and try to monitor the remote machine.		☑
Set Up Topology	Open topology window and define your environment. You can choose one of the templates.	Topology	☑
Set Up Element Properties	For each element define server name, application and version.	Topology	☐

9

Creating and Configuring Execution Steps

You use Execution session steps to check a server's performance and monitor measurements.

When you create a session from a cartridge, the session includes a number of Execution steps. You may need to create additional Execution steps to include in your session, to suit your particular environment and needs.

You can create new Execution steps from scratch, or create them by importing existing LoadRunner scenarios. This chapter describes how to create and configure Execution session steps.

This chapter discusses:

- Creating New Execution Session Steps
- Importing LoadRunner Scenarios
- Specifying Step Execution Order
- Managing Scripts
- Setting an Initial Load (Manual Profiles)
- Configuring Script Details
- Using Relative Paths for Scripts

About Execution Session Steps

Each Execution step performs a specific business process and contains one or more scripts. LoadRunner Tuning Module provides a set of *canned* or

prepared scripts for the execution steps. These scripts include tests for the Web Infrastructure.

Note: For detailed information on the provided canned scripts, see "Canned Scripts" on page 90.

You can use the canned scripts, or write your own scripts and incorporate them in the execution steps.

You can also create Execution steps by importing LoadRunner scenarios.

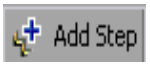
You select one or more scripts for each execution step. You can also indicate upon which servers to execute the steps. When selecting multiple scripts, you can specify whether the scripts should all be part of one session step or contained in separate steps. Since LoadRunner Tuning Module executes single steps, combining scripts in a step causes them to be executed together.

Vusers are virtual users that emulate real users. After setting up the steps for your session, you can indicate the number of Vusers you want to emulate. You can also choose a goal for each session step. For more information, see Chapter 12, "Scheduling Session Steps."

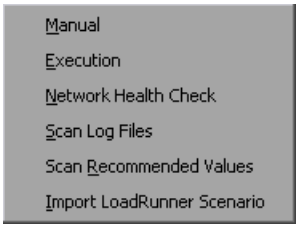
Creating New Execution Session Steps

Creating a new Execution step involves choosing and configuring scripts.

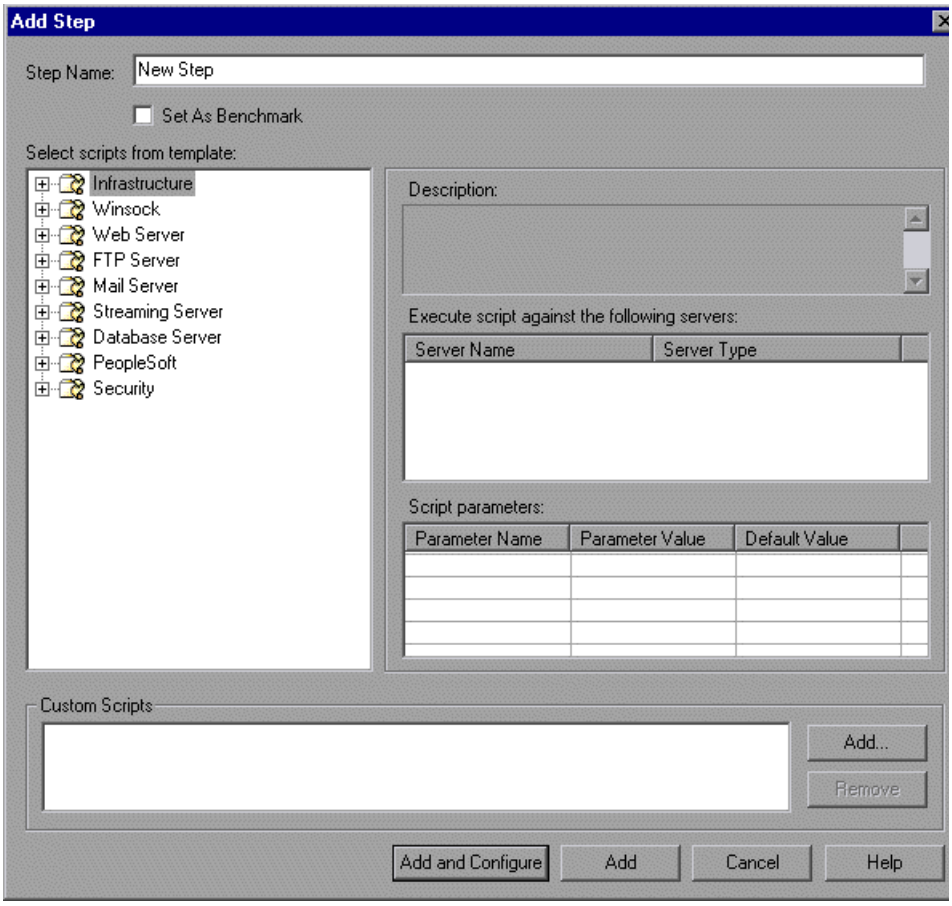
To create a new Execution step:



1 Click **Add Step** or choose **Session** > **Add Step**. The step type menu is displayed.



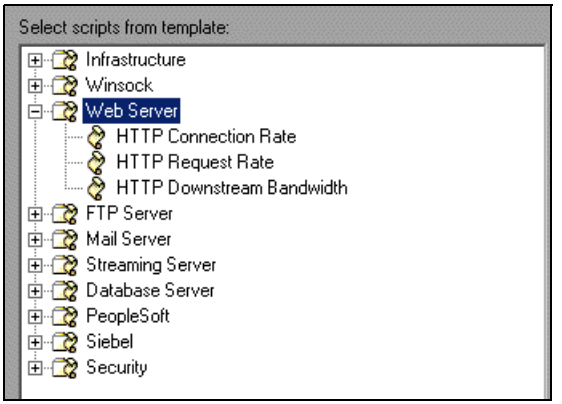
2 Choose **Execution** to open the Add Step dialog box.



3 In the Step Name field, enter a name for the step or accept the default name.

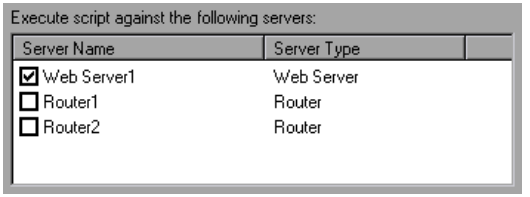
4 To specify that this step should be used as the benchmark step, check **Set As Benchmark**. The benchmark step is executed whenever you click **Execute Benchmark Step** in the Session tab.

5 In the Select Scripts from Template section, expand the script type that you want to use (if it is not already expanded). The left pane lists the types of canned scripts. Clicking a type expands it and shows all of its scripts. For example, clicking Web Server shows you the following scripts: HTTP Connection Rate, HTTP Request Rate and HTTP Downstream Bandwidth.

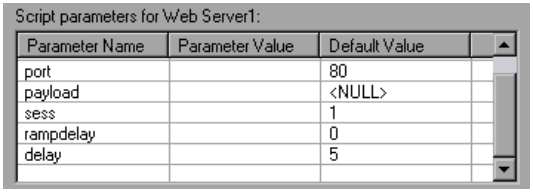


6 Click the script that you want your step to execute. Note that the Description section displays information about the selected script.

7 In the **Execute script against the following servers** section, LoadRunner Tuning Module displays the servers against which you can run the script. To specify the servers against which you want the script to run, check the boxes adjacent to their names.



8 In the Script Parameters section, LoadRunner Tuning Module displays a table containing the arguments for the selected script and their default values.



To change a value, click the appropriate cell in the Parameter Value column and enter the new value.

Note: A command line parameter is limited to one line.

---

9 To add a custom script to a session step, click the **Add...** button to the right of the Custom Scripts pane, browse to the script that you want to use, and click **Open**. The script name is added to the Custom Scripts pane.

10 Repeat steps 6 through 9 for all the scripts and servers that you want to add to your step.

11 When you have finished choosing the scripts for your step, you have two choices:

➤ Click the **Add** button below the Custom Scripts pane.

This closes the Add Step dialog box. The new Execution step is displayed in the session tree.

➤ Click **Add and Configure**. The Add Step dialog box is closed, the new step is displayed in the session tree, and the Step Configuration dialog box opens. Configure the step with the appropriate settings, and click **Close**. The new Execution step is added to the session tree.

Understanding the Add Step Dialog Box

You use the Add Step dialog box to insert a new Execution step into the session. You create the step by choosing from the canned scripts or adding custom scripts created with the Virtual User Generator.

You can assign several scripts to each executable session step. You can also indicate upon which servers to execute these steps.

Set as Benchmark

Defines this step as the benchmark step. This is the step that LoadRunner Tuning Module executes when you click **Execute Benchmark Step** in the Session tab. You typically run the benchmark step before and after tuning, to measure the improvement in your system's performance.

Selecting Scripts

You select the scripts to include in your step by choosing from the list of canned scripts and choosing custom-written scripts:

➤ **Select scripts from template:** The left pane lists the types of canned scripts. Clicking a type expands it and shows all of its scripts. Select the canned script you want to add to your session step.

➤ **Execute script against the following servers:** Displays the servers to which the selected canned script applies and can therefore be executed. To specify the servers against which to run the script, check the boxes adjacent to those servers.

➤ **Add...** (to the right of the Custom Scripts pane): Adds a custom script to a session step. Click the button, browse to the script that you want to use, and click **Open**. The script name is added to the Custom Scripts pane.

➤ **Remove:** Removes a selected custom script from the Custom Scripts pane.

➤ **Add** (below the Custom Scripts pane)**:** Closes the Add Step dialog box and adds the new step to the session tree.

➤ **Add and Configure:** Adds the new step to the session tree, and then opens the Step Configuration dialog box.

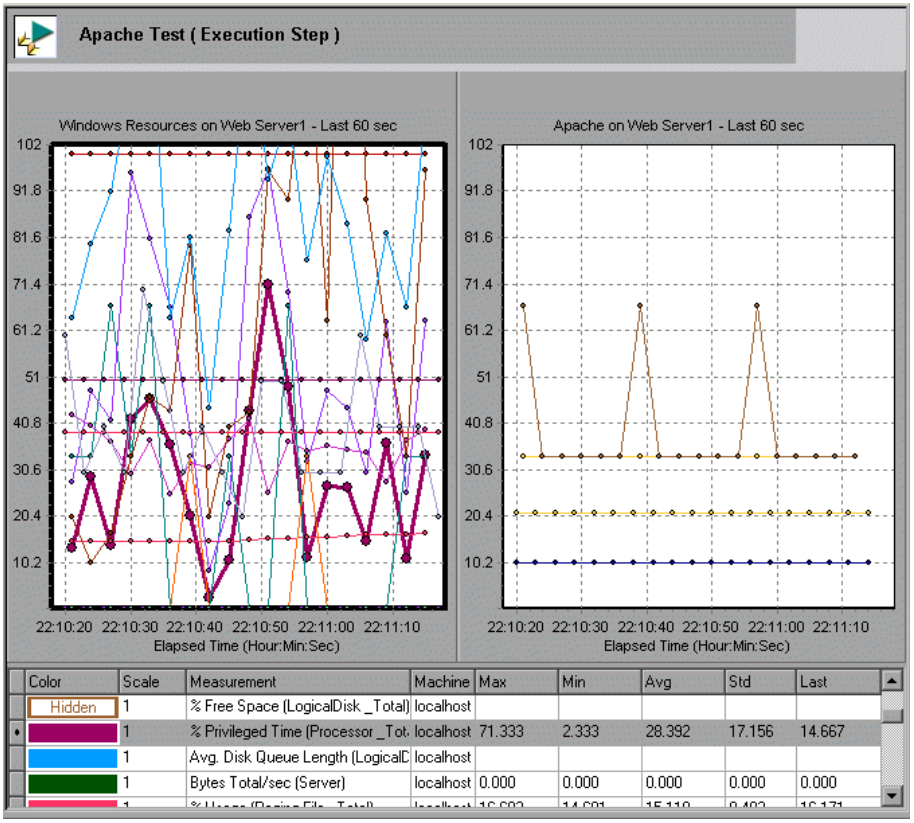
Description: The name and description of the selected script.

Script parameters: Displays the arguments for the selected script and their default values. You can specify your own value in the **Parameter Value** column. To specify a value, click the column in the row of the argument.

Tip: To view information about a parameter, hold your cursor over the parameter name and view the tool tip.

Execution Step View

When you create an Execution step, or when you select an existing step in the session tree, the right side of the Session tab shows graphs of the measurements that the step is monitoring.



The toolbar includes the following buttons for controlling and configuring the step:

Remove Step: Deletes the selected step from the session.

Move to the Next Step: Changes the focus to the step below the current step in the session tree.

Execute Benchmark Step: Executes the step defined as the benchmark step. See “Understanding the Add Step Dialog Box” on page 61.

Configure: Opens the Step Configuration dialog box. See "Understanding the Step Configuration Dialog Box," on page 67.

Scheduler: Opens the Schedule Builder window. See "Scheduling Session Steps," on page 119.

Run Step: Executes the step.

Gradual Stop: Gradually stop the step.

Stop: Stop the step immediately.

Add Vusers: Opens the Run/Stop Vusers dialog box.

View and Control Vusers: Opens the Vuser Management dialog box.

Add/Remove Graphs: Opens the Select Online Graphs dialog box. See "Understanding the Select Online Graphs Dialog Box" on page 298.

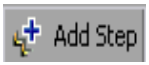
Description: This section is used for information about the session, step or script, depending on which is currently selected in the session tree. You can edit it at any time. The text that you enter is saved when you save the session.

Importing LoadRunner Scenarios

You can import existing LoadRunner scenarios to create session steps.

Note: You can import goal-oriented scenarios and manual scenarios in the Percentage Mode. If you need to import a scenario created in the Vuser Group Mode, first convert it (using LoadRunner Controller) to the Percentage Mode (**Scenario > Convert Scenario to the Percentage Mode**) and then import it.

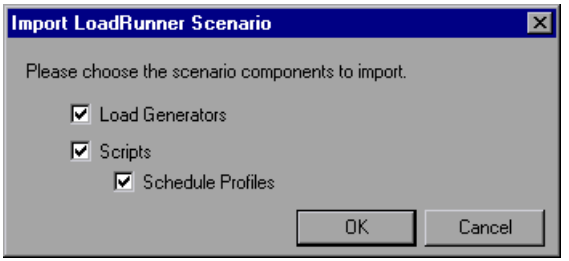
To import a scenario:



1 Click **Add Step** or choose **Session** > **Add Step**. The step type menu is displayed.

2 Choose **Import LoadRunner Scenario**. The Open Scenario dialog box is displayed.

3 Locate the scenario that you want to import (a file with the *.lrs* extension) and click **Open**. The Import LoadRunner Scenario dialog box is displayed.



4 Check the appropriate boxes and then click **OK**.

LoadRunner Tuning Module creates a step using the scenario's settings, and adds it to the session tree.

Understanding the Import LoadRunner Scenario Dialog Box

You use the Import LoadRunner Scenario dialog box to specify which parts of the scenario to import. Check the appropriate boxes:

Load Generators: Check this box to import the scenario's load generators. The load generators are added to the list in the Load Generators window (**Session** > **Load Generators**).

Scripts: Check this box to import the scenario's scripts. The scripts will be attached to the step that is created by importing the scenario.

Schedule Profiles: Check this box to import all the scenario's schedule profiles. The profiles are added to the Profile Name box in the Schedule Builder window, and the session step is assigned the profile that was assigned to the scenario. If you leave this box clear, the step is assigned the default profile: if the scenario is goal-oriented, the step is assigned the

default goal-oriented profile; if manual, it is assigned the default manual schedule. **Note:** If you do not import scripts, this box is disabled.

Specifying Step Execution Order

When you create an Execution step, you can specify which subsequent step should execute after the current step completes execution. This allows you to define a methodology: if the step succeeds, you can specify that Step A should subsequently be executed; if it fails, the user should execute Step B. You can then specify subsequent steps for Step A and Step B, and so on.

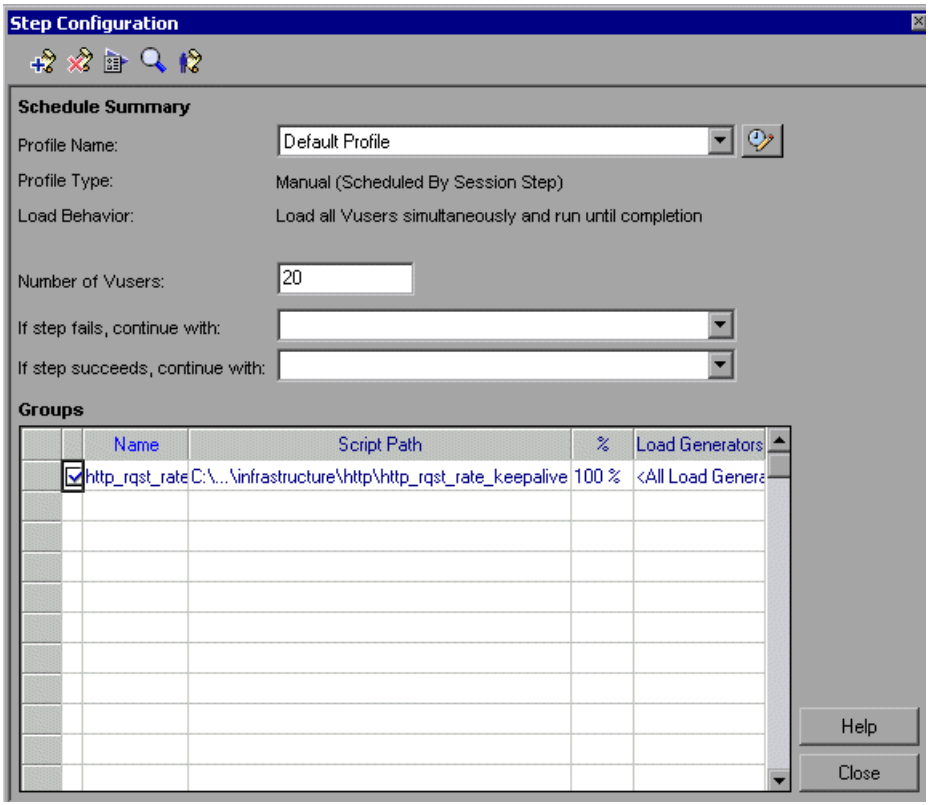
For the purpose of determining which step should subsequently execute, a failed step is one that ends when an alert causes all the Vusers to halt execution. If the step ends in any other way, it is considered to be a successful step.

If you do not specify a subsequent step (for success, failure, or both), execution stops when the current step has completed.

To specify subsequent steps:

1 Create the subsequent steps (see "Creating New Execution Session Steps," on page 57).

2 In the Session tab, click **Configure**. The Step Configuration dialog box is displayed.



3 In the **If step fails...** and **If step succeeds...** boxes, choose or type the names of the steps that should be executed if the current step fails or succeeds, respectively. (Optional)

4 Click **Close**.

Understanding the Step Configuration Dialog Box

You use the Step Configuration dialog box to configure Execution steps.

Add Script: Opens the Add Script dialog box, allowing you to add a canned or custom script to your step.

Remove Script: Removes the selected script from the step.

Run-Time Settings: Opens the Run-Time Settings dialog box, enabling you to edit the script run-time settings you previously set using VuGen.

Script Details: Opens the Script Information dialog box, displaying details about the selected script.

Modify the Vuser script Opens the Virtual User Generator, so that you can edit the script. For more information on editing scripts, refer to the *Creating Vuser Scripts* guide.

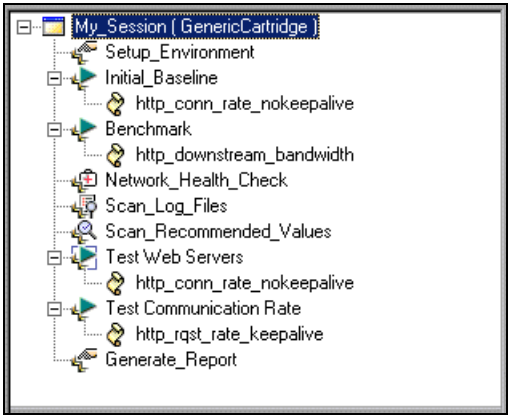
The Schedule Summary section includes the following fields:

- **Profile Name:** To choose another profile, select one from the Profile Name list box.
- **Profile Type:** Manual or Goal-Oriented.
- **Load behavior:** See "Defining Load Behavior," on page 136.
- **Number of Vusers** (manual profiles only)**:** To change the initial number of Vusers in the step, enter the new value here.
- **If step fails...:** Choose or type the name of the step that should be executed if the current step fails. (Optional)
- **If step succeeds...:** Choose or type the name of the step that should be executed if the current step completes successfully. (Optional)

Scheduler: Opens the Schedule Builder dialog box, allowing you to edit the profile settings. See "Creating and Selecting a Profile," on page 121.

Managing Scripts

After you add scripts to your Execution steps, you can add, delete, or view them from the step tree in the left pane of the Session tab.



Adding a Script to a Step

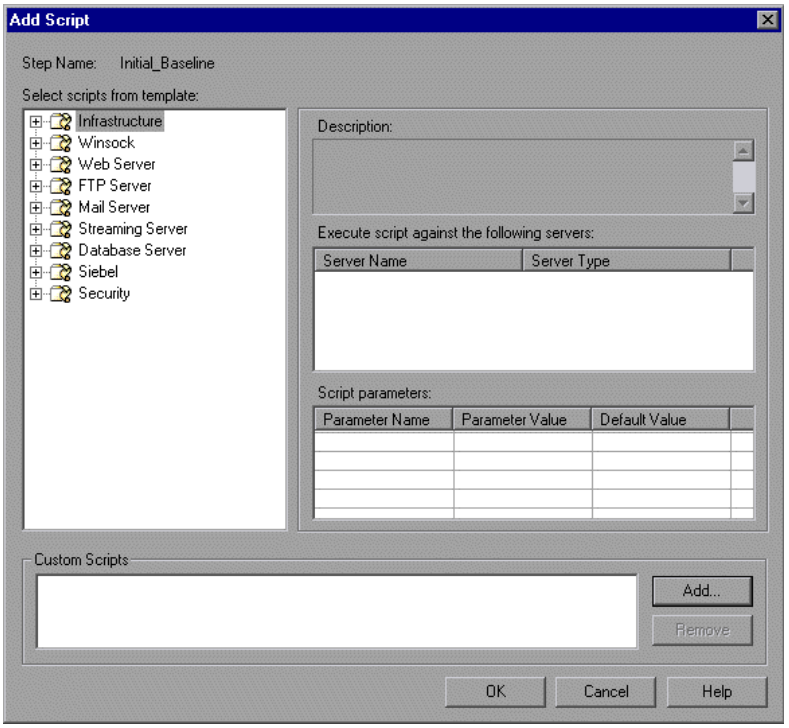
You can add canned or custom scripts to Execution steps.

To add a script to a step:

1 Select the step to which you want to add the script.

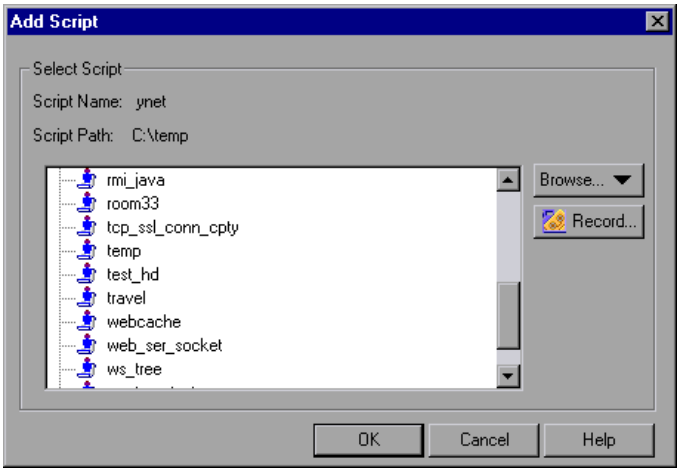
2 Click **Configure** to open the Step Configuration dialog box, and then click **Add Script**. Alternatively, you can right-click the step and choose **Add Script**.

The Add Script dialog box is displayed.



3 To add a canned script to your step, choose it from the tree in the left pane of the dialog box, and configure its settings. For details on how to choose and configure a canned script, see "Creating New Execution Session Steps," on page 57.

4 To add a custom script to your step, click **Add** in the Add Step dialog box. The Add Custom Script dialog box is displayed.



5 To add an existing custom script, select one of the displayed scripts and click **OK.** To change the path, click **Browse** and select an alternate path. To record a new script in the LoadRunner Tuning Module Virtual User Generator, click **Record**. For more information about recording scripts, refer to the *Creating Vuser Scripts* guide.

6 Click **OK** to close the Add Step dialog box. The script is added to the step and appears in the session tree.

Understanding the Add Custom Script Dialog Box

The Add Custom Script dialog box enables you to add scripts to your execution step.

Select Script: Displays the available scripts in the current directory.

➤ **Script Name:** Click the script that you want to add to your session step. The script appears in the Script Name column.

➤ **Script Path:** Displays the script directory's path.

➤ **Browse:** Select a script from a different directory.

Note: To select a VB Vuser script, browse to locate the .usr file.

➤ **Record:** Opens the Virtual User Generator so that you can begin recording a script. For more information on recording scripts, see the *Creating Vuser Scripts* guide.

Note: While a step is running, you can add Vuser scripts to the session step and enable them. However, if you add a script after all the Vusers in the session step have been ramped up, the new script will not run in the step.

Deleting, Viewing and Modifying Scripts

To delete a script from a step:

1 Select the step from which you want to delete the script.

2 Click **Configure** to open the Step Configuration dialog box, and then click **Remove Script**. Alternatively, you can right-click the script in the session tree and choose **Remove Script**.

To view or modify a script:

1 Select the script.

2 Click **Modify the Vuser script**. LoadRunner Tuning Module opens the Virtual User Generator, so that you can edit the script. For more information on editing scripts, refer to the *Creating Vuser Scripts* guide.

Working with Groups

The Step Configuration dialog box includes a table of all the scripts assigned to a step.

To view information about a step's scripts:

1 Select the step in the session tree.

2 Click **Configure** to open the Step Configuration dialog box. The Groups section shows the script information.

Groups

	Script Name	Script Path	%	Load Generators
☑	web_ser_socket	R:\LR_TESTS\web_ser_socket	14.29 %	<All Load Generators>
☐	test_hd	R:\test_hd	14.29 %	<All Load Generators>
☐	ws_tree_test	C:\temp\ws_tree_test	14.29 %	localhost
☑	browserlevel	R:\LR_TESTS\browserlevel	14.29 %	<All Load Generators>
☑	tcp_conn_cpty	E:\...\network\tcp_conn_cpty\tcp_conn_cpty	14.29 %	<All Load Generators>
☑	webcache	R:\LR_TESTS\webcache	14.29 %	<All Load Generators>
☑	tcp_ssl_conn_cpty	E:\...\network\tcp_ssl_conn_cpty\tcp_ssl_conn_cpty	14.29 %	<All Load Generators>

You can perform the following actions from this window:

Sort the scripts: To sort the scripts by their name, path, percentage or Load Generator, click the title of the desired column.

Enable a Script: Check the box adjacent to the script name. When you execute the step, all of the enabled scripts are executed.

Disable a Script: Clear the box adjacent to the script name. When you execute the step, none of the disabled scripts are executed.

Rename a Script: Double-click an item in the **Script Name** column and enter the new name.

Add a Script to the List: Click in the next available row of the **Script Name** column, and click on the arrow to the right of the box. A list box opens showing all the scripts in the most recent path. To add a script from the list, select the script and click **OK.** To change the path, click **Browse** and select an alternate path.

Modify the Vuser Percentage: When you assign multiple scripts to a step, LoadRunner Tuning Module runs them simultaneously. By default, LoadRunner Tuning Module distributes the scripts evenly between the Vusers. For example, if you assign two scripts to one step, fifty percent of the Vusers run one script, while the remaining fifty percent run the other. To modify the script distribution, click in the **%** column and modify the percentages. Note that the total sum of the percentages must equal 100.

To add a load generator:

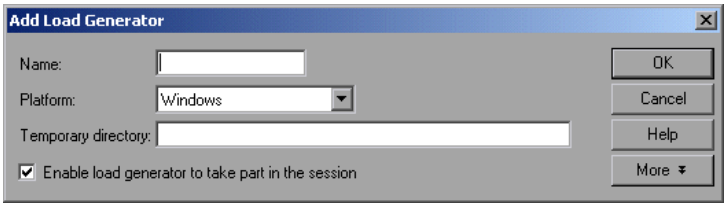
1 The Load Generators column automatically contains <All Load Generators> for each script. You can assign specific load generators for each script. Click in the **Load Generators** column in the next available row.

2 Click the arrow to the right of the box. A list box opens showing the available load generator machines.

3 Select one or more machines and click **OK**.

4 Click **All Generators** to instruct LoadRunner Tuning Module to run the script on all available machines.

5 To add a new load generator, click the first entry in the list box, **Add**. The Add Load Generator dialog box opens:



6 Type the name of the load generator in the **Name** box.

7 In the **Platform** box, select the type of platform on which the load generator is running.

8 By default, LoadRunner Tuning Module stores temporary files on the load generator during session execution, in a temporary directory specified by the load generator's TEMP or TMP environment variables. To override this default for a specific load generator, type a location in the **Temporary Directory** box.

9 To allow the load generator to take part in the session, check **Enable load generator to take part in the session**.

10 Click **OK** to close the Add Load Generator dialog box. LoadRunner Tuning Module adds the new load generator to the Load Generator Name list.

To include the new load generator in your session, select it from the Load Generator Name list, and click **OK**. Note that you can select multiple load generators.

Repeat the above procedure for each load generator you want to add to your session step.

For more information about setting up load generators, see Chapter 11, "Managing Load Generators."

To configure a load generator:

➤ Use the Load Generators dialog box to set a load generator's attributes while adding it to the load generator list, or to modify the attributes of an existing load generator at any time.

➤ You can also use the Load Generators dialog box to indicate which load generators will run Vusers in the session step. For example, if a load generator is unavailable for a particular session step, you can use the Load Generators dialog box to exclude it temporarily instead of removing it entirely from your list of load generators. For instructions on using the Load Generators dialog box, see "Configuring Load Generators" on page 101. To configure additional load generator settings, see "Configuring Load Generator Settings" on page 107.

➤ To configure global settings for all load generators participating in the session, use LoadRunner Tuning Module's Options dialog box. For more information, see Chapter 14, "Configuring Session Steps."

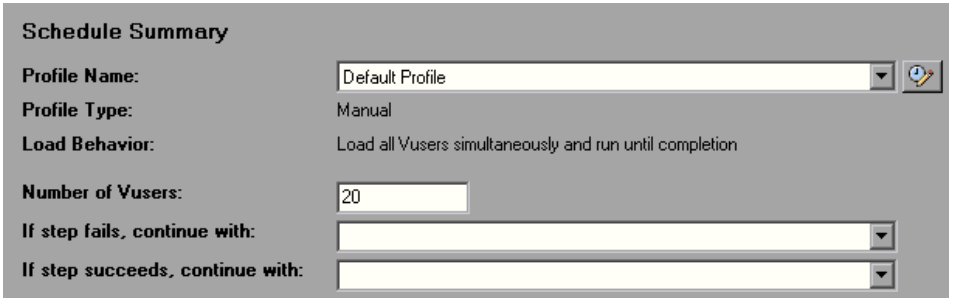
Setting an Initial Load (Manual Profiles)

If the profile associated with a step is a manual one, you can specify the number of Vusers that will execute the step.

To define an initial load:

1 Select the session step in the session tree, and click **Configure**. The Step Configuration dialog box is displayed. The Schedule Summary section shows the following fields:

- Profile Name
- Profile Type (Manual or Goal-Oriented)
- Load behavior
- Number of Vusers



2 In the Number of Vusers field, enter the number of Vusers that should execute the step. This value is also used when you delay the start of step execution using the Session Start dialog box (**Session > Start Time**)—LoadRunner Tuning Module runs the session step at the designated time with the specified number of Vusers.

Note that you can change the number of Vusers while a step is executing, via the Run/Stop Vusers dialog box. See "Adding Vusers to a Running Execution Step" on page 196.

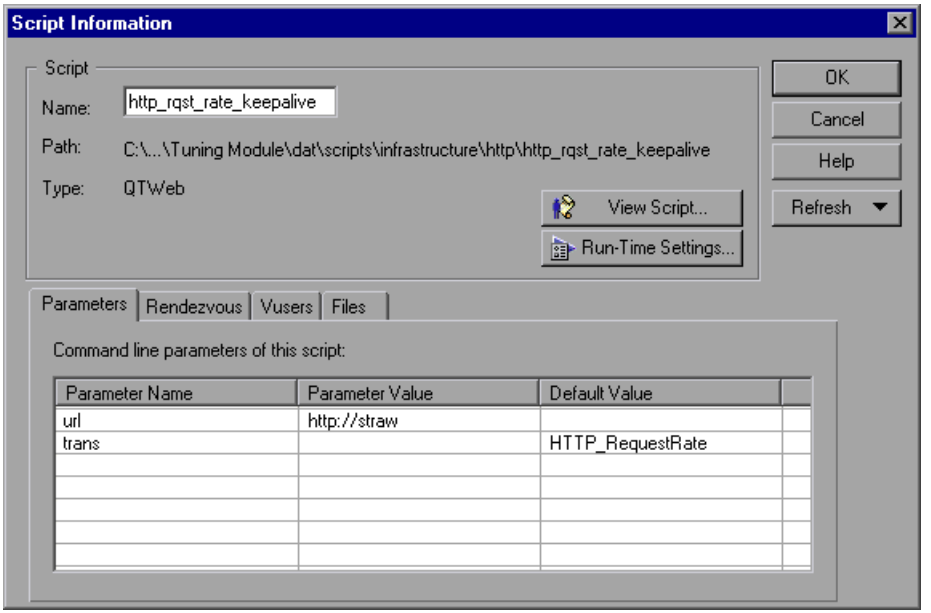
Configuring Script Details

You can view the details of a script, edit the script, or change its run-time settings.

To view script details:

1 Select the step or script in the session tree and click **Configure**. The Step Configuration dialog box is displayed. The Groups section lists all the scripts used by the step.

2 Select a script and click the **Script Details** button. The Script Information dialog box opens, displaying details about the selected script.



Note: Another way to display the Script Information dialog box is to right-click the script in the session tree and then choose **Script Details**.

---

3 Make any necessary changes to the script details.

4 Click **OK** to close the Script Information dialog box.

After you add steps and scripts to your session, you can configure the run-time options and set a schedule. Refer to "Typical Workflow" on page 11 for a summary of the testing procedure. After you set up your session steps, you run the steps and begin your tuning session. For information on running the tuning session, see Chapter 17, "Running an Execution Step."

Understanding the Script Information Dialog Box

The Script Information dialog box lets you view the details of a selected script and modify its settings. It also displays the script's command line options (for example, -url Application Server1), and tabs displaying information about parameters, rendezvous points, Vusers and files.

Script: Displays details of the selected script.

- **Name:** Displays the name of the selected script. To modify the name, type the modified name in the Name box.
- **Path:** Displays the script directory's path.
- **Type:** Displays the type of the selected script.
- **View Script:** Opens the Virtual User Generator, so that you can edit the script. For more information on editing scripts, see the *Creating Vuser Scripts* guide.

Note: If you use VuGen to make changes to a script while the Console is running, click the **Refresh** button and select **Script** to update the script details in the session step.

- **Run-Time Settings:** Opens the Run-Time Settings dialog box, enabling you to edit the script run-time settings you previously set using VuGen. If you did not set run-time settings for a script in VuGen, the default VuGen settings are displayed for all but the Log and Think Time tabs, which display the default Console settings. Note that several protocols, such as Web and Java, have specific settings. For information on the run-time settings, see the VuGen Help or the *Creating Vuser Scripts* guide.

Note: If you modify the run-time settings from the Console, LoadRunner Tuning Module runs the script using the modified settings. To restore the initial settings, click the **Refresh** button and select **Run-Time Settings**.

➤ **Parameters Tab:** Displays the command line parameter names, parameter values, and default values for the selected script. To specify your own value, click in the **Parameter Value** column and enter the desired value.

Tip: To view information about a parameter, hold your cursor over the parameter name and view the tool tip.

➤ **Rendezvous Tab:** Displays the rendezvous points defined for the selected script.

➤ **Vusers Tab:** Displays all Vusers associated with the selected script. If you have not created Vusers yet, the box is empty.

➤ **Files Tab:** Displays a list of all the files used by the script, and allows you to add a file to the list. By default this list shows all files in the script's directory (only after your script has been added to the script list). These files include the configuration settings file, the init, run, and end portions of the script, the parameterization definitions file, and the *usr* file. Note that you can delete the files that you add, but not the other files listed. To exclude a file from the list, select the check box adjacent to it. To add a file to the list, click **Add**.

Refresh: Click this button and select **Script** to update the script details in the session step, if you make any changes to a script while the Console is running. Select **Run-Time Settings** to restore the initial run-time settings, if you modify the run-time settings from the Console. **Note**: Refreshing the Log run-time settings does not override the **Send messages only when an error occurs** option, even if the original setting was **Always send messages**. Refreshing the Think Time run-time settings does not override the **Replay think time** option, even if the original setting was **Ignore think time**.

Using Relative Paths for Scripts

To specify the location of a script, you can either browse to the script or type its relative location into the Script Path column. The location can be relative to the current session directory, or the LoadRunner Tuning Module installation directory.

You can specify a path relative to the current session directory by typing either of the following notations at the start of the script path:

.\	indicates that the path is relative to the location of the session directory.
..\	indicates that the path is relative to the location of the parent directory of the session directory.

For example, if the current session is located at *F:\sessions*, to specify a script located at *F:\sessions\scripts\user1.usr*, you could type the following:

```
.\scripts\user1.usr
```

You can specify a path relative to the LoadRunner Tuning Module installation directory by typing a percent sign (%) at the beginning of the script path. For example, if the LoadRunner Tuning Module installation directory is located at *F:\LoadRunner Tuning Module*, to specify a script located at *F:\LoadRunner Tuning Module\scripts\user1.usr*, you could type the following:

```
%\scripts\user1.usr
```

Note: When specifying a relative path, you can include standard DOS notation (.\ and ..\) inside the path, as shown in the following example: M:\LRALT\my_tests\..\..\test.usr.

When you run a session, by default, the script is copied to a temporary directory on the Vuser group machine. This enables the Vuser group load generator to access the script locally instead of over a network.

You can instruct the Console to store the script on a shared network drive (see Chapter 14, "Configuring Session Steps.") If you configure the Console to save the script to a network drive, you must ensure that the Vuser load generator recognizes the drive. The Script window contains a list of all the scripts and their paths. A script's path is based on the Console load generator's mapping of that location. If a Vuser load generator maps to the script's path differently, path translation is required. Path translation converts the Console load generator's mapping to the Vuser load generator's mapping. For more information see Appendix D, "Performing Path Translation."

10

Using Built-In Knowledge

This chapter describes the Cartridge and canned script features. It describes the contents of a cartridge, shows how to modify its contents, and how to save it. It also lists and explains the canned scripts.

This chapter discusses:

- Cartridge Features
- Saving Cartridges
- Editing Auto Assign Settings
- Canned Scripts

About Using Cartridges and Canned Scripts

Cartridges provide pre-packaged tuning methodologies for specific application environments (for example, Siebel, PeopleSoft or J2EE environments). The pre-defined session steps save you time in designing tuning sessions, and the packaged methodology guides you through the process.

You can use the supplied cartridges without changing them, or customize them according to your particular requirements.

When you create a session, you base it on a cartridge. You can then save the session, or save it as a cartridge.

Canned scripts are predefined tests that you can use to test your system components.

Cartridge Features

A cartridge includes the following features:

- A comprehensive workflow that guides you through an iterative process to tune your specific environment
- Pre-defined alerts and fuses that speed up the process of identifying performance bottlenecks and help to prevent system crashes
- Recommended configuration settings for infrastructure elements (such as load balancers)
- Pre-created business processes
- Pre-defined monitor settings for your specific application and environment

You can further customize a cartridge for your specific environment. A customized cartridge also gives you a workflow that allows you to quickly diagnose production performance problems, and lets you measure performance when you upgrade your application or the underlying infrastructure.

Saving Cartridges

When you save a session as a cartridge, the following information is included in the cartridge:

- **Topology templates:** The list of templates that you can use for creating your topology. If you create new templates or delete existing ones, the saved cartridge will contain the updated list. See "Creating Topology Diagrams from Templates," on page 31.
- **Canned scripts:** The canned scripts that appear in the session. These can vary, depending on the cartridge on which you based the session. See "Adding Session Steps," on page 48.
- **Settings for auto-assigned monitors:** The measurements that are tracked when you use the Auto Assign feature to assign monitors. For example, if you add a measurement to the Apache monitor, the new measurement will also be tracked when you Auto Assign the Apache monitor to a server. See "Accessing the SiteScope Administration Console" on page 46.

- **Settings for auto-assigned alerts:** The alerts that are assigned when you use the Auto Assign feature. See "Accessing the SiteScope Administration Console" on page 46.
- **Updated recommended values:** If you change the recommend values for a service, the changed values are saved as the cartridge's recommended values. For details of how to change recommended values, see "Changing Recommended Values," on page 680.
- **Rules for log file scanning:** The rules that determine which errors from the log files are displayed when you perform the Scan Logs step. See "Scan Logs Step," on page 227.
- **Session steps and workflow:** The session steps and the order in which they appear in the session.

To create a cartridge from a session:

1. Choose **Cartridge** > **Save Cartridge** or **Cartridge** > **Save Cartridge As**. The Save Cartridge dialog box is displayed.
2. Enter the cartridge name and click **Save**. A new cartridge is created.

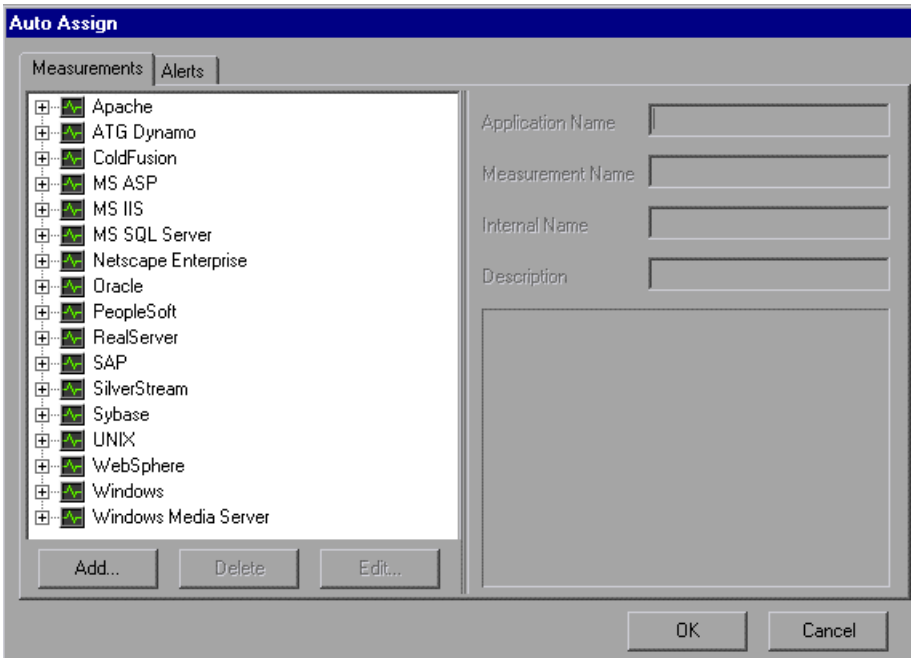
Editing Auto Assign Settings

Each cartridge has its own customized settings for auto assignment of monitors and alerts. These include the measurements that each monitor tracks, and the alert conditions and actions. You can change these settings so they reflect your particular needs—you can add or delete measurements, and change alerts.

To add measurements to a monitor, you import them from a previously-saved session in which the measurements were assigned to the monitor. For example, to add an Apache Web server measurement, you import it from another session where that measurement was assigned to the Apache Web server monitor.

To edit auto assignment settings:

1 Choose **Cartridge** > **Edit Auto Assign**. The Edit Auto Assign dialog box is displayed.



2 Use the Measurement tab to add, delete or edit monitor measurements.

3 Use the Alerts tab to configure alerts.

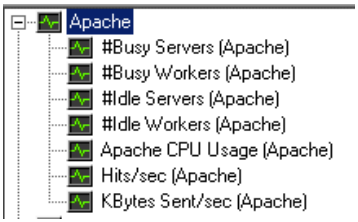
Understanding the Edit Auto Assign dialog box

You use the Edit Auto Assign dialog box to change the list of measurements and alerts that are applied to a server when you use the Auto Assign feature.

Measurements

The Measurements tab displays all the available monitors, and lets you add, delete or edit monitor measurements. To display the monitor measurements

that are used when the Auto Assign function is applied to a server, expand the monitor's element.



Add: Click to add a measurement to the monitor list. See "Adding Measurements," on page 88.

Delete: Click to delete a measurement from the monitor list. See "Deleting Measurements," on page 90.

Alerts

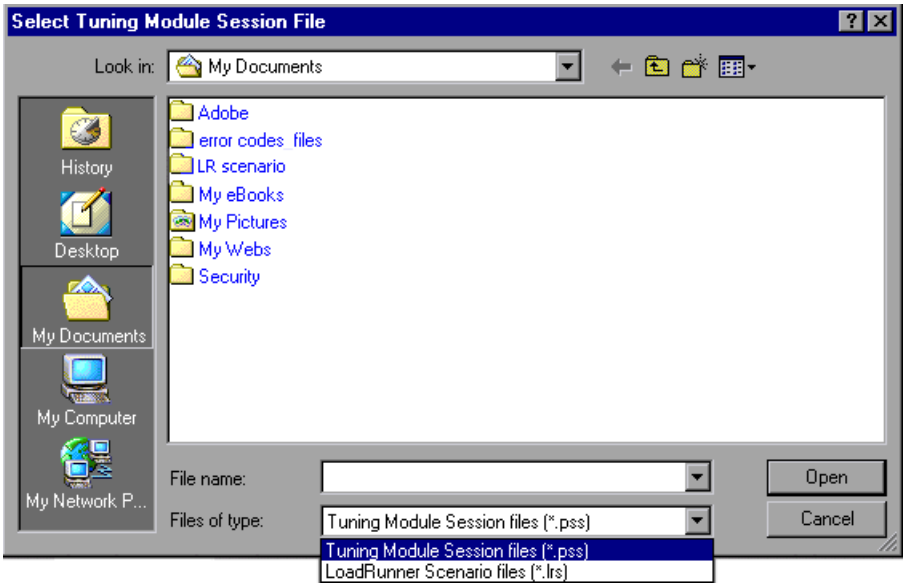
The Alerts tab displays all the available alerts, the conditions that trigger them, the actions they cause, and the relevant troubleshooting information. You can change the list of alerts that are applied by the Auto Assign feature. See "Editing Alerts," on page 90.

Adding Measurements

You use the Measurements tab to add auto-assigned measurements to monitors.

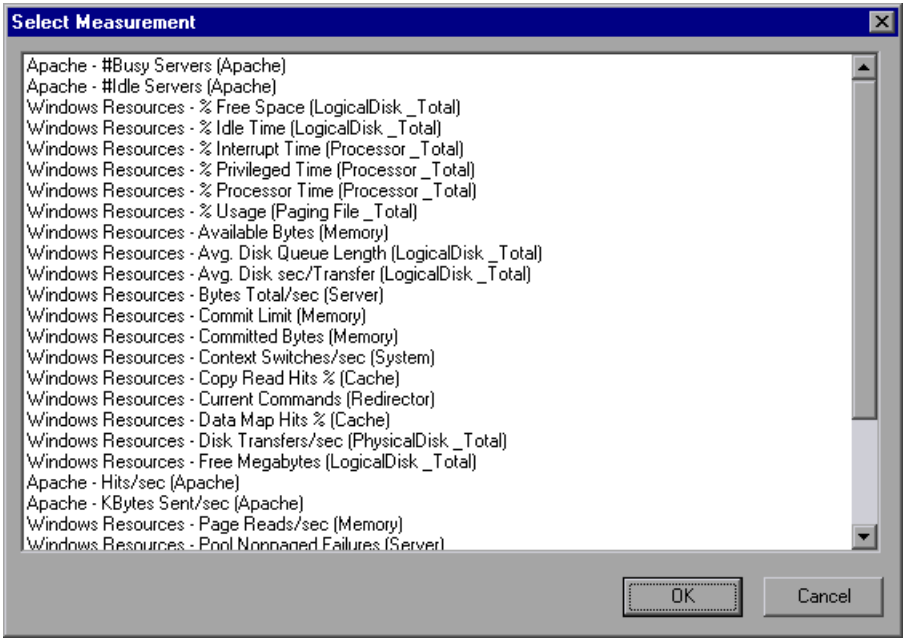
To add a measurement to a monitor:

1 Click **Add**. The Select Session File dialog box is displayed.



2 You now need to open the session file that contains the monitor measurements you want to add. This can be a file with either a *.pss* or a *.lrs* extension. Locate the file and click **Open**.

The Select Measurement dialog box is displayed. This dialog box shows you all the measurements assigned to monitors in the session file (that is, all the measurements that were assigned to monitors in the session).



3 Select the measurement that you want to add, and click **OK**. The Edit Auto Assign dialog box appears, with the measurement you chose added to the list of measurements for the monitor.

Note: When you add WebLogic, WebSphere & PeopleSoft (TUXEDO) measurements, you should replace any specific server name in the measurement name with "<SRVNAME>". When you add WebSphere measurements, you should also replace any specific node in the measurement name with "<NODE>".

Deleting Measurements

To delete a measurement from a monitor:

➤ In the Measurements tab, select the measurement and click **Delete**. The measurement is removed from the monitor's list of measurements.

Editing Alerts

To edit the alerts that are applied when you use the Auto Assign feature:

➤ Use the Alerts tab to add, delete, or configure alerts. For information on using this tab, see "Defining Alerts," on page 139.

Canned Scripts

LoadRunner Tuning Module's canned (built-in) scripts help you test the connection capacity or rate goals. You can use the canned scripts to create session steps in the following areas:

➤ Infrastructure

➤ Winsock

➤ Web Server

➤ FTP Server

➤ Mail Server

➤ Streaming Server

➤ Database Server

➤ PeopleSoft

➤ Siebel

➤ Security

Infrastructure

The Infrastructure scripts relate to the infrastructure of your network architecture. The available script is:

DNS Request Rate: Checks the rate at which host names are resolved on the Domain Name Server. The script title as it appears in the script list is *dns_rqst_rate*. The supported server is the DNS server. **Note:** The DNS protocol is not supported by UNIX.

Winsock

The Winsock scripts relate to the TCP/IP and SSL connections of your servers. The available scripts are:

TCP Connection Capacity: Sustains simultaneous TCP connections with the target device. The script title as it appears in the script list is *tcp_conn_cpty*. The supported servers are Web servers, load balancers and database servers.

SSL Connection Capacity: Sustains simultaneous SSL connections with the target device. The script title as it appears in the script list is *tcp_ssl_conn_cpty*. The supported servers are Web servers, load balancers and database servers.

Web Server

The Web Server scripts are used for testing Web server performance. The available scripts are:

HTTP Connection Rate: Generates HTTP requests against a target URL, without keep-alive enabled. Therefore, each new request must establish a new connection with the Web server. The script title as it appears in the script list is *http_conn_rate_nokeepalive*. The supported servers are Web servers and load balancers.

HTTP Request Rate: Generates HTTP requests against a target URL, with keep-alive enabled. Therefore, each new request reuses the same connection with the Web server. Note: the Web server must configured to support keep-alive connections in order for this script to work properly. The script title as it appears in the script list is *http_rqst_rate_keepalive*. The supported servers are Web servers and load balancers.

HTTP Downstream Bandwidth: Generates downstream data transmission by a continuous download of a large file from the Web server via HTTP protocol. The HTTP requests are set with keep-alive enabled. Therefore each new request re-uses the same connection with the web server. The script title as it appears in the script list is *http_downstream_bandwidth*. The supported servers are Web servers and load balancers.

HTTPS Connection Rate: Generates HTTPS requests against a target URL, without keep-alive enabled. Therefore, each new request must establish a new connection with the Web Server. The script title as it appears in the script list is *https_conn_rate_nokeepalive*. The supported servers are Web servers and load balancers.

HTTPS Request Rate: Generates HTTPS requests against a target URL, with keep-alive enabled. Therefore, each new request reuses the same connection with the Web Server. The script title as it appears in the script list is *https_rqst_rate_keepalive*. The supported servers are Web servers and load balancers.

Note: the Web Server should be configured to support keep-alive connections in order for this script to work properly.

FTP Server

The FTP Server scripts are used for testing FTP server performance. The available scripts are:

FTP Connection Capacity: Sustains simultaneous FTP connections with the file server. The script title as it appears in the script list is *ftp_conn_cpty*. The supported servers are FTP servers, Web servers and load balancers.

FTP Get File Rate: Generates FTP GET requests to download a specific file from an FTP server. The script title as it appears in the script list is *ftp_get_rate*. The supported servers are FTP servers, Web servers and load balancers.

FTP Put File Rate: Generates FTP PUT requests to upload a specific file to an FTP server. The script title as it appears in the script list is *ftp_put_rate*. The supported servers are FTP servers, Web servers and load balancers.

Mail Server

The Mail Server scripts are used for testing the following mail protocols: SMTP, MAPI, POP3 and IMAP. The available scripts are:

SMTP Connection Capacity: Sustains simultaneous SMTP connections with the mail server. The script title as it appears in the script list is *smtp_conn_cpty*. If a connection is closed by the server, it is re-established in order to sustain the same number of sessions. The supported servers are Mail Servers and Application Servers.

SMTP Send Mail: Submits e-mail messages to a mail server via the SMTP protocol. The script title as it appears in the script list is *smtp_send_mail*. The supported servers are Mail Servers and Application Servers.

MAPI Connection Capacity: Sustains simultaneous MAPI sessions with MS Exchange Server. These sessions are created only once. If the session terminates, it is not re-established. The script title as it appears in the script list is *mapi_conn_cpty*. The supported servers are Mail Servers and Application Servers.

MAPI Send Mail: Sends an e-mail with a file attachment to the specified recipient, using the given MS Exchange profile. The script title as it appears in the script list is *mapi_send_mail*. The supported servers are Mail Servers and Application Servers.

POP3 Connection Capacity: Creates simultaneous POP3 session connections with the mail server. If a mail server does not allow enough connections, it will reject new attempts to connect, and performance will suffer. The purpose of this script is to test the maximum number of connections that the specified mail server allows. The script title as it appears in the script list is *pop3_conn_cpty*. The supported servers are Mail Servers and Application Servers.

POP3 Retrieve Mail: Puts stress on the infrastructure involved in retrieving mail messages on a POP3 server. The script title as it appears in the script list is *pop3_retrieve_mail*. The supported servers are Mail Servers and Application Servers.

IMAP Connection Capacity: Sustains simultaneous IMAP sessions with the mail server. The script title as it appears in the script list is *imap_conn_cpty*. The supported servers are Mail Servers and Application Servers.

IMAP Search Mail: Searches for mail in the specified folder using the given search criteria. The script title as it appears in the script list is *imap_search_mail*. The supported servers are Mail Servers and Application Servers.

IMAP Store Mail: Stores a mail message in the specified mail folder using the IMAP protocol. The *imap_mail.dat* file is an IMAP message supplied with LoadRunner Tuning Module. To use a different message, modify or replace this file. The script title as it appears in the script list is *imap_store_mail_file*. The supported servers are Mail Servers and Application Servers.

Streaming Server

The Streaming Server scripts are used for testing streaming servers that support the Real and Microsoft Media Stream protocols. The available scripts are:

Real Connection Capacity: Sustains simultaneous connections through the RTSP protocol with the Real server. The script title as it appears in the script list is *rtsp_conn_cpty*. The supported servers are Web Servers and Application Servers.

RealPlayer Play Media: Plays a media stream through the RTSP protocol. The script title as it appears in the script list is *rtsp_play_media*. The supported servers are Web Servers and Application Servers.

MMS Play Media: Plays a media stream via the Microsoft Media Stream (MMS) protocol. The script title as it appears in the script list is *mms_play_media*. The supported servers are Web Servers and Application Servers.

Note: Before running this script in the context of a load test, make sure that your Streaming Media Server public directory includes the file wmload.asf (this is a Microsoft requirement), or, if you are not running in the context of a load test, add the mms_disable_host_check() function to your script.

Database Server

The Database Server scripts are used for testing database servers that support the ODBC driver and ADO-DB interface. The available scripts are:

ADO-DB Connection Rate: Opens a new connection to the SQL server, executes the specified query on the server and closes the connection. The script title as it appears in the script list is *adodb_open_sql_close_rate*. The supported servers are Database Servers.

ADO-DB SQL Query Rate: Executes the specified SQL statement on the server. The connection is created only once at the script initialization stage. The script title as it appears in the script list is *adodb_sql_query_rat*e. The supported servers are Database Servers.

ODBC SQL Connection Rate: Executes the specified SQL statement on the database server. A new connection is opened in each iteration. The script title as it appears in the script list is *odbc_sql_connection_rate*. The supported servers are Database Servers.

ODBC SQL Query Rate: Executes the specified SQL statement on the database server. The connection to the database server is created only once at the script initialization stage. The script reuses the same connection for all of its iterations, but creates a new database cursor for each iteration. The script title as it appears in the script list is *odbc_sql_query_rate*. The supported servers are Database Servers.

ODBC SQL Query Rate (reuse cursor): Executes the specified SQL statement on the database server. The connection to the database server is created only once at the script initialization stage. The script reuses the same connection and database cursor for all of its iterations. The script title as it appears in the script list is *odbc_sql_query_rate_reuse_cursor.* The supported servers are Database Servers.

Oracle Connection Rate: Opens a connection to a server via SQL*Plus, holds the connection for a predefined period (default: 1 second), and then closes it. The script uses the Oracle 2-tier protocol. The script title as it appears in the script list is *oracle_connection_rate*. **User input:** server name, user name, password, delay.

Oracle Connection Capacity: Opens a connection to a server via SQL*Plus, and holds it by running a query once every 30 seconds. The script uses the

Oracle 2-tier protocol. The default query is "select user from dual". Each user creates a connection. The script title as it appears in the script list is *oracle_connection_capacity.* **User input:** server name, user name, password, SQL query.

Oracle SQL Rate: Opens a connection to a server via SQL*Plus, and runs a query, and waits for 1 second (configurable). The script uses the Oracle 2-tier protocol. The script title as it appears in the script list is *oracle_sql_rate.* **User input:** server name, user name, password, SQL query, delay.

PeopleSoft

The PeopleSoft scripts are generalized business processes used to create load on most of the application's components. Scripts are available for PeopleSoft 8.1x and PeopleSoft 8.4x.

The available scripts for PeopleSoft 8.1x are:

Financials Process Financial Information: Browses the Process Financial Information tree in the PeopleSoft8.1x Financials application, using the following schema:

Process financial Information > Review Financial Information > Inquire > Journal

The PeopleSoft login and logout are not included in the iteration loop.

Financials Browse Admin Procurement: Browses the Administer Procurement tree in the PeopleSoft8.1x Financials application, using the following schema:

Administer Procurement > Create Payments > Inquire > Voucher Inquiry

The PeopleSoft login and logout are not included in the iteration loop.

Financials Browse Manage Assets: Browses the Manage Assets tree in the PeopleSoft 8.1x Financials application, using the following schema:

Manage Assets > Manage Assets > Use > Asset ExpressAdd

The PeopleSoft login and logout are not included in the iteration loop.

Financials Browse Manage Sales Activities: Browses the Manage Sales Activities tree in the PeopleSoft 8.1x Financials application, using the following schema:

Manage Sales Activities > Maintain Customers > Use > General Information

The PeopleSoft login and logout are not included in the iteration loop.

The available scripts for PeopleSoft 8.4x are:

CRM Browse FieldService Service Orders: Browses the FieldService tree in the PeopleSoft 8.4x CRM application, using the following schema:

FieldService > Service Orders

The PeopleSoft login and logout are not included in the iteration loop.

CRM Browse FieldService Reports: Browses the FieldService tree in the PeopleSoft 8.4x CRM application, using the following schema:

FieldService > Reports > Expense Report

The PeopleSoft login and logout are not included in the iteration loop.

CRM Browse Sales Leads: Browses the Sales tree in the PeopleSoft 8.4x CRM application, using the following schema:

Sales > Leads > Lead Details

The PeopleSoft login and logout are not included in the iteration loop.

CRM Browse Sales Opportunities: Browses the Sales tree in the PeopleSoft 8.4x CRM application, using the following schema:

Sales > Opportunities > Opportunity Details

The PeopleSoft login and logout are not included in the iteration loop.

PeopleSoft Login: Performs login and logout.

PeopleSoft Ping: Monitors the PS Ping page. **Note:** You can use this script with PeopleTools 8.42 and later.

Siebel

The Siebel scripts are generalized business processes used to create load on most of the application's components. The available script is:

Siebel 7.x Portal User Login: Logs on to the Siebel 7.x portal, and immediately logs off.

Siebel 7 SWSE Test: Tests the Siebel Web Server Extension (SWSE). The script sends a request for the Siebel 7.x login page but does not perform an actual login. This generates a load on the Web server and on the SWSE plug-in. User input: server name, application name, user name, password. The script is only available in the Siebel Cartridge.

Security

The Security scripts simulate Denial of Service (DoS) attacks on servers.

Note: Refer to the documentation on each script to see the platforms on which it can be run. For information on viewing Denial of Service data, see "Security Monitoring," on page 637.

Note: Before you can use a LoadRunner load generator (as opposed to a LoadRunner Tuning Module load generator) to execute a DDoS script, you need to make some changes on the load generator machine. For details, see "Running DDoS Scripts on LoadRunner Load Generators" on page 747.

The available scripts are:

Bonk DoS Attack: Disrupts the sequence of IP packet fragments by changing the information in the fragment's header information. Large IP packets are frequently broken up into fragments when they pass through routers on the Internet. The fragment's header includes an offset field that specifies the fragment's position in the packet. The attacker creates a sequence of fragments with offset fields greater than the header length. If

the destination host cannot handle the overlapping fields, it crashes, freezes or reboots. The script title as it appears in the script list is *ddos_bonk*. **Note:** This script can be run only on a load generator running Windows XP.

Ping of Death DoS Attack: Creates a packet whose size is too great for the destination host to handle. The TCP/IP specification allows creating a packet with a maximum size of 65536 bytes. The attacker uses the Ping utility to create an IP packet with a size greater than 65536 bytes. This can cause the destination system to crash, freeze, or reboot. The script title as it appears in the script list is *ddos_ping_of_death*. **Note:** This script can be run only on a load generator running Windows XP.

SYN Flood DDoS Attack: Causes the server's queue to overflow by filling it with half-open (also known as *pending*) connections. To do this, it initiates a TCP/IP connection, using IP spoofing to specify an unreachable or down client as the sender. The server mistakenly identifies each TCP/IP connection as a genuine one coming from a different IP address, and responds by sending a SYN/ACK signal to the client. It then waits for an ACK signal (which never arrives) to complete the three-way handshake. The connection is stored in the server's queue, which keeps growing as the attacker creates more half-open connections, until it eventually overflows, preventing the server from receiving any new connections. Although the connections eventually expire, the attacker can initiate new fake connections at a rate faster than the server can cause them to expire. The script title as it appears in the script list is *ddos_syn_flood*. **Note:** Can be run only on machines running Windows 2000 or Windows XP.

Targa3 DoS Test: Sends combinations of uncommon IP packets to hosts to generate attacks using invalid fragmentation, protocol, packet size, header values, options, offsets, TCP segments, routing flags, and other unknown or unexpected packet values. This script is useful for testing IP stacks, routers, firewalls, NIDs (Network Intrusion Detectors) and other similar components for stability and reactions to unexpected packets. Some of these packets might not pass through routers with filtering enabled; tests with the source and destination hosts on the same Ethernet segment give the best effects. The script title as it appears in the script list is *ddos_targa3*. **Note:** This script can be run only on a load generator running Windows 2000 or Windows XP.

Teardrop DoS Attack: Disrupts the sequence of IP packet fragments by changing the information in the fragment's header information. Large IP

packets are frequently broken up into fragments when they pass through routers on the Internet. The fragment's header includes an offset field that specifies the fragment's position in the packet. The attacker creates a sequence of fragments with offset fields that overlap. If the destination host cannot handle the overlapping fields, it crashes, freezes or reboots. The script title as it appears in the script list is *ddos_teardrop*. **Note:** This script can be run only on a load generator running Windows XP.

UDP Echo DDoS Attack: Sends a UDP datagram to the destination address. In particular, this illustrates the danger of having the UDP echo service turned on in */etc/inetd.conf* on many versions of UNIX. Consider the result if the source address and port are set to localhost and 7 respectively: the inetd in FreeBSD 2.2 seems to detect this denial of service attack, but many UNIX variants do not. The script title as it appears in the script list is *ddos_udp_echo*. **Note:** Can be run only on machines running Windows 2000 or Windows XP.

11

Managing Load Generators

After choosing the tests you want to run, and the components that you want to test, you need to specify the computers from which LoadRunner Tuning Module will run the tests. These computers are called load generators. This chapter describes how to define and manage load generators.

This chapter discusses:

- Configuring Load Generators
- Configuring Load Generator Settings

About Managing Load Generators

You use load generator machines for running the tests on your components. When you assign a script to a step, you also specify the load generators that will run the script, and can configure their properties.

Configuring Load Generators

You can set a load generator's attributes while adding it to the load generator list, or modify the attributes of an existing load generator at any time, using the Load Generators dialog box.

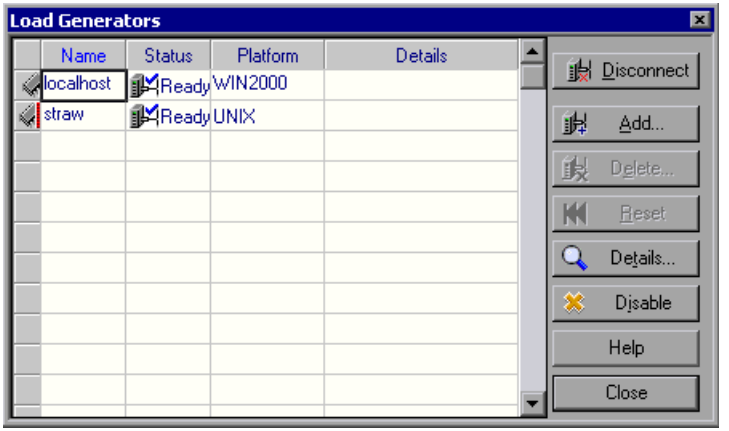
To configure global settings for all load generators participating in the session, use the Timeout tab in the **Tools** > **Options** dialog box. For more information, see Chapter 14, "Configuring Session Steps." To set properties specific to each load generator, use the Load Generators dialog box as described below.

You can also indicate which load generators will run Vusers in the session. For example, if a load generator is unavailable for a particular session run, you can exclude it temporarily instead of removing it entirely from your list of load generators.

You select which load generators will take part in the session by using the Enable and Disable commands. Disabling a load generator temporarily removes it from the list. Enabling a load generator reinstates it. Disabling load generators is particularly useful if you want to isolate a specific machine to test its performance.

To configure a load generator:

1 Click the **Generators** button, or select **Session** > **Load Generators**. The Load Generators dialog box opens. For each load generator, the **Name**, **Status**, **Platform**, and **Details** are displayed.



2 Click the load generator to select it, and click the relevant buttons to configure it.

Understanding the Load Generators Dialog Box

The Load Generators dialog box displays information about the load generators connected to the session step.

Name: Lists the name of the load generator.

Status: Displays the status of the load generator. The following table describes the possible statuses of the load generator.

Status	Description
Ready	The load generator is connected
Connecting	The load generator is in the process of connecting
Active	The load generator is running Vusers
Down	The load generator is not connected
Failed	A connection with the load generator could not be established

Platform: Displays the type of platform on which the load generator is running.

Details column: In the event that the connection fails, displays details about why it failed.

Connect: Instructs the Console to connect the load generator for the session step. The load generator's status changes from DOWN to READY. When the load generator is connected, the button automatically changes to **Disconnect**. Click **Disconnect** to change the load generator's status from READY to DOWN.

Add: Opens the Add Load Generator dialog box.

Delete: Deletes the load generator. The load generator can only be deleted when it is disconnected.

Reset: Attempts to reset a failed connection.

Details button: Opens the Load Generator Information dialog box to display information about the selected load generator. See "Understanding the Load Generator Information dialog Box," on page 107.

Disable/Enable: Instructs the Console to disable or enable the selected load generator. To enable or disable a load generator, select it and click Enable or Disable, respectively. When a load generator is disabled, its Name, Status, Platform, and Details appear in grey. When a load generator is enabled, they appear in blue.

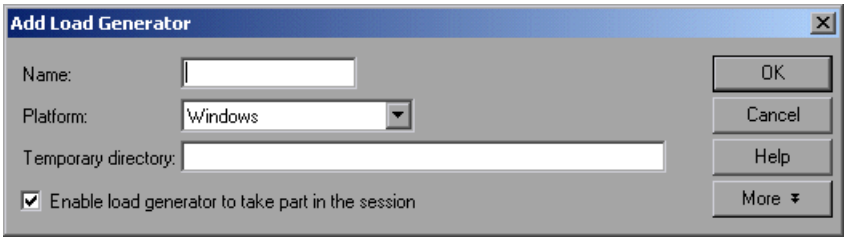
Note: The Console monitors a Windows load generator machine's CPU usage and automatically stops loading Vusers on a load generator when it becomes overloaded. You can monitor the status of a machine's CPU usage using the icons in this dialog box. When the CPU usage of a load generator becomes problematic, the icon to the left of the load generator name contains a yellow bar. When the machine becomes overloaded, the icon contains a red bar.

Adding a Load Generator

You can add load generator machines to your session step, or modify information for existing load generators.

To add a load generator, or modify load generator information:

1 To add a load generator, or modify information for an existing load generator, click **Add**. The Add Load Generator dialog box opens.



2 Specify the load generator machine and its platform.

3 To allow the load generator to take part in the session, check **Enable load generator to take part in the session**.

4 Click **OK** to close the Add Load Generator dialog box.

5 To remove a load generator, select it and click **Delete**.

6 Click **Close** to close the Load Generators dialog box. The load generator name you entered appears in the Load Generators list; its status is set to Down.

Note: The LoadRunner Tuning Module Console monitors a Windows load generator machine's CPU usage and automatically stops loading Vusers on a load generator when it becomes overloaded. You can monitor the status of a machine's CPU usage using the icons in the Load Generators dialog box. When the CPU usage of a load generator becomes problematic, the icon to the left of the load generator name contains a yellow bar. When the machine becomes overloaded, the icon contains a red bar.

Understanding the Add New Load Generator Dialog Box

You can add load generator machines to your session step using the Add New Load Generator dialog box.

Name: Type the name of the load generator you want to add in the Name box.

Platform: Select the type of platform on which the load generator is running.

Temporary Directory: A location on the load generator machine where the Console can store temporary files. By default, LoadRunner Tuning Module stores temporary files on the load generator during session step execution, in a temporary directory specified by the load generator's TEMP or TMP environment variables. The location that you enter in this field overrides the default for a specific load generator.

Enable load generator to take part in the session: Select to include the load generator in the session.

More: Expands the dialog box and shows the following additional tabs where you can configure load generator settings:

- **Status**
- **Run-Time Quota**
- **Firewall**
- **Run-Time File Storage**

- **Unix Environment**
- **Vuser Limits**

Note: For information on configuring these settings, see "Configuring Load Generator Settings" on page 107.

Configuring Load Generator Settings

You can configure additional settings for individual load generators using the tabs in the Add Load Generator or Load Generator Information dialog boxes. The settings that can be configured are: Run-Time File Storage, UNIX Environment, Run-Time Quota, Vuser Limits, Connection Log (Expert mode), and Firewall.

You can configure global settings for all load generators participating in the session, using the Options dialog box. For more information, see Chapter 14, "Configuring Session Steps."

To configure load generator settings:

1. From the Add Load Generator or Load Generator Information dialog box, click **More** to expand the box and show the Status, Run-Time File Storage, UNIX Environment, Run-Time Quota, Vuser Limits, and Firewall (when the load generator is not the localhost) tabs. You can also configure the Connection Log Settings while working in Expert mode. For information on the Connection Log tab, see "Working in Expert Mode" on page 759.
2. The settings apply to the load generator specified in the Name box. To configure a load generator other than the one specified, enter the name and platform of the load generator in the Name box, or select the load generator from the Load Generator dialog box.
3. Select the tab that contains the load generator settings you want to configure, and configure the settings.
4. Click **OK** to close the Add New Load Generator or Load Generator Information dialog box and save your settings.

Understanding the Load Generator Information dialog Box

The Load Generator Information dialog box enables you to add a load generator machine to your session step.

Name: Type the name of the load generator you want to add in the Name box.

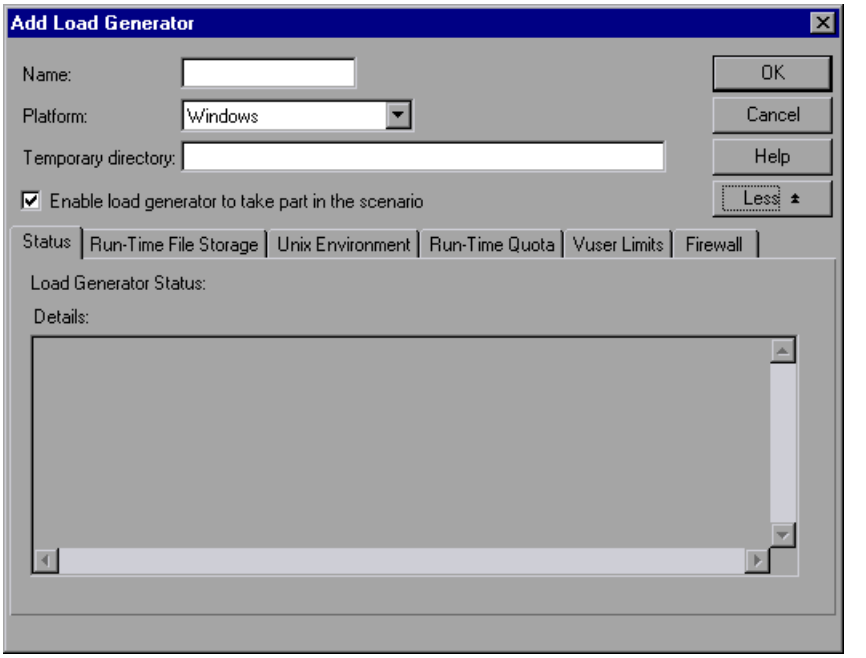
Platform: Select the type of platform on which the load generator is running.

Temporary directory: Type a location, on the load generator, where the Console can store temporary files.

Enable load generator to take part in the session: Select to include the load generator in the session step.

Status Tab

Select the **Status** tab to display details of the Load Generator Status

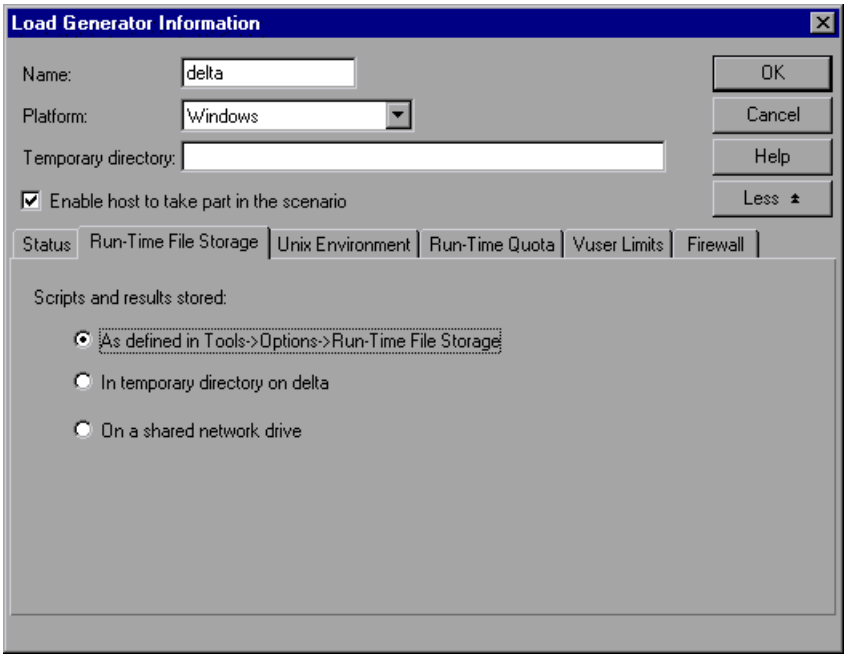


Load Generator Status: Displays the status of the load generator.

Details: Displays error and other run-time information about the selected load generator.

Run-Time File Storage Tab

Select the **Run-Time File Storage** tab to specify the result directory for the performance data that LoadRunner Tuning Module gathers from each load generator during a session.



Scripts and results stored: Select one of the following options:

- **As defined in Tools > Options> Run-Time File Storage:** Stores the results as specified in the global settings.
- **In temporary directory on <*load generator name*>:** Instructs the Console to save the run-time files (results of the step run and Vuser scripts) on a hard drive of the load generator computer.
- **On a shared network drive:** Instructs the Console to save the step results and/or the Vuser scripts on a shared network drive. A shared network drive is a drive to which the Console and all the load generators in the step have read and write permission.

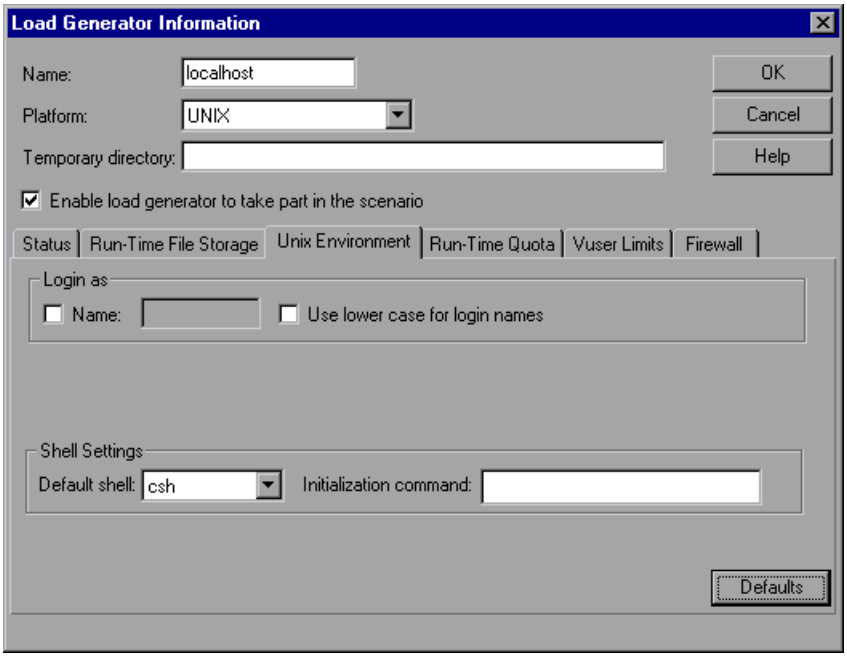
Note: If the load generator is *localhost*, LoadRunner Tuning Module stores the scripts and results on a shared network drive, and the checkboxes and radio buttons for setting the location are all disabled.

If you are monitoring over the firewall, the Run-Time File Storage settings are not relevant.

To set the network location for the results, see Chapter 16, "Preparing to Run a Session Step."

UNIX Environment Tab

Select the **UNIX Environment** tab to configure the login parameters and shell type for each UNIX load generator.



Login as

- **Name:** If the load generator is UNIX-based, set the login information for the load generator. By default, LoadRunner Tuning Module uses your NT user name for the UNIX login. In other words, if your NT login is *ptune*, the Console will log on to the load generator as *ptune*. To log on to a UNIX-based load generator using a different login name, select the **Name** check box and specify the desired UNIX login name. Using this option you can log on to the NT Console as *bill* and connect to the UNIX load generator as *mike*. However, you should make sure that *mike* allows *bill* to log on using his name. This can be done by adding the line "+ bill" at the beginning of mike's *.rhosts* file.
- **Use lower case for login names:** Instructs LoadRunner Tuning Module to use lower case names during login to avoid case-sensitive issues with the UNIX operation system.

Note: For information on the Local User setting available in Expert mode, see "Working in Expert Mode" on page 759.

Shell Settings: Specify the UNIX shell settings for the remote UNIX load generator.

- **Default shell:** Select the default shell on the UNIX load generator: **csh** (C Shell—the default), **bsh** (Bourne Shell), or **ksh** (Korn Shell).

Note: To allow LoadRunner Tuning Module to run your application under the Korn shell, you first need to make sure that the *.profile* file contains all of the LoadRunner Tuning Module environment settings—for example, the M_LROOT definition and the LicenseManager variable. These environment settings already exist in your *.cshrc* file. Your UNIX $M_LROOT/templates directory contains a template for the *.profile* file, called *dot profile*. Use the template as a guide for modifying your *.profile* file with the LoadRunner Tuning Module environment settings.

In addition, if you are using a Korn shell (ksh), you must delete all LoadRunner Tuning Module settings from the *.cshrc* file (e.g. M_LROOT) before executing the session.

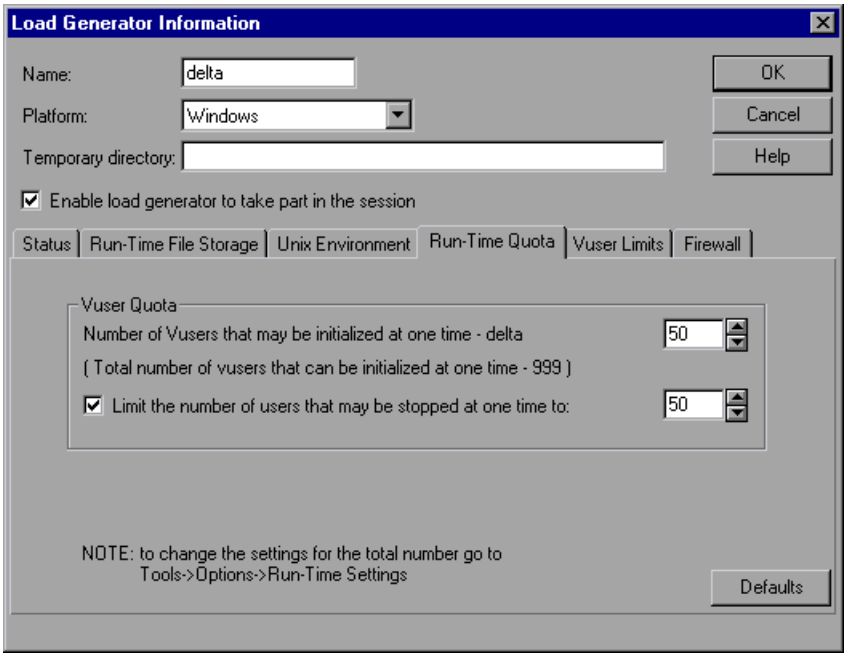
➤ **Initialization command:** Enter any command line options for LoadRunner Tuning Module to use when logging on to a UNIX system. This initialization command will run as soon as the shell opens. For example, you could select *ksh* and use the following initialization command:
. .profile;

Note: If you are monitoring or running Vusers over the firewall, the UNIX Environment settings are not relevant.

Run-Time Quota Tab

Initializing or stopping a large number of Vusers simultaneously places large stress on a load generator. To reduce stress on a load generator, you can initialize or stop smaller batches of Vusers.

Select the **Run-Time Quota** tab to specify the maximum number of Vuser types that the load generator will initialize or stop simultaneously.



Vuser Quota

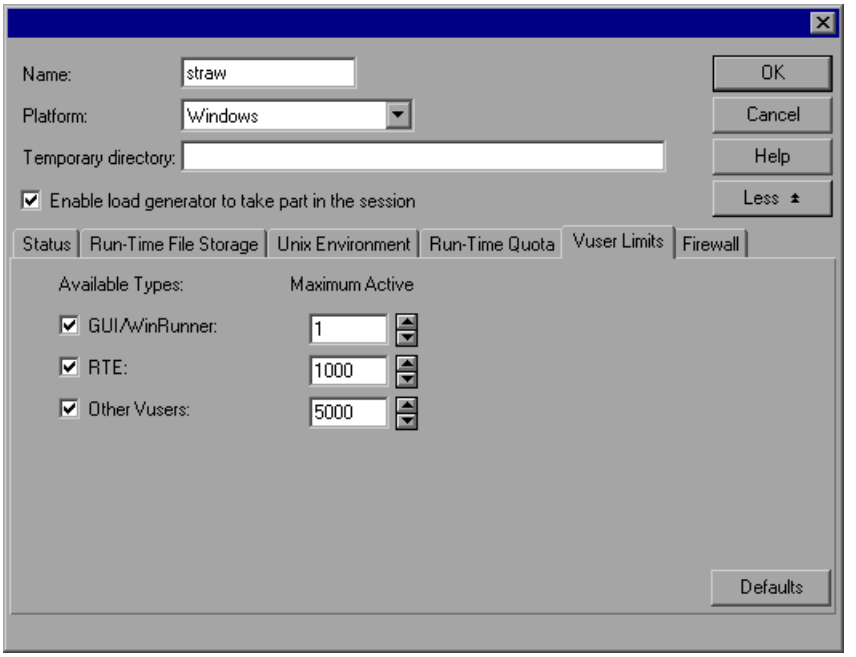
- **Number of Vusers that may be initialized at one time - current load generator:** Select the maximum number of Vusers that the current load generator can initialize simultaneously.
- **Limit the number of users that may be stopped at one time to:** Select the maximum number of Vusers that the current load generator can stop simultaneously.

Defaults: Sets the number of Vusers that may be initialized or stopped at one time to 50.

You can set run-time quotas for an entire session using the Run-Time Settings tab in the Options dialog box. For information on setting quotas globally for an entire session, see Chapter 14, “Configuring Session Steps.”

Vuser Limits Tab

Select the **Vuser Limits** tab to modify the maximum number of GUI, RTE, and other Vusers that a load generator can run.



Available Types: Select the type(s) of Vusers you want the load generator to run.

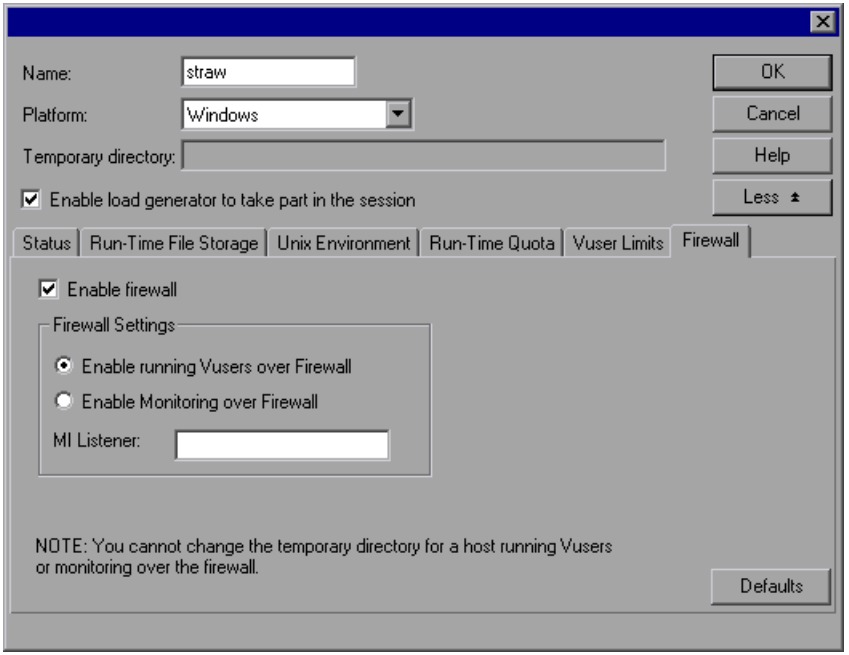
Maximum Active: Select the maximum number of Vusers of each type for the load generator to run.

Defaults: Sets GUI-WinRunner to 1, RTE to 1000, and Other Vusers to 5000.

Note: The maximum number of active Vusers that you specify must not exceed the number of Vusers that you are licensed to run. To check your Vuser licensing limitations, choose **Help > About LoadRunner Tuning Module**.

Firewall Tab

Select the **Firewall** tab to enable monitoring or running Vusers through a firewall.



Enable firewall: Enables LoadRunner Tuning Module to monitor or run Vusers over a firewall.

Note: If you select the Enable Firewall option, the Temporary directory option for storing temporary files is disabled. Any location in the Temporary directory box is erased.

Firewall Settings:

- **Enable Monitoring over Firewall:** Enables LoadRunner Tuning Module to monitor the load generator machine through a firewall.
- **Enable running Vusers over Firewall**: Enables LoadRunner Tuning Module to run Vusers on a load generator machine outside the firewall.

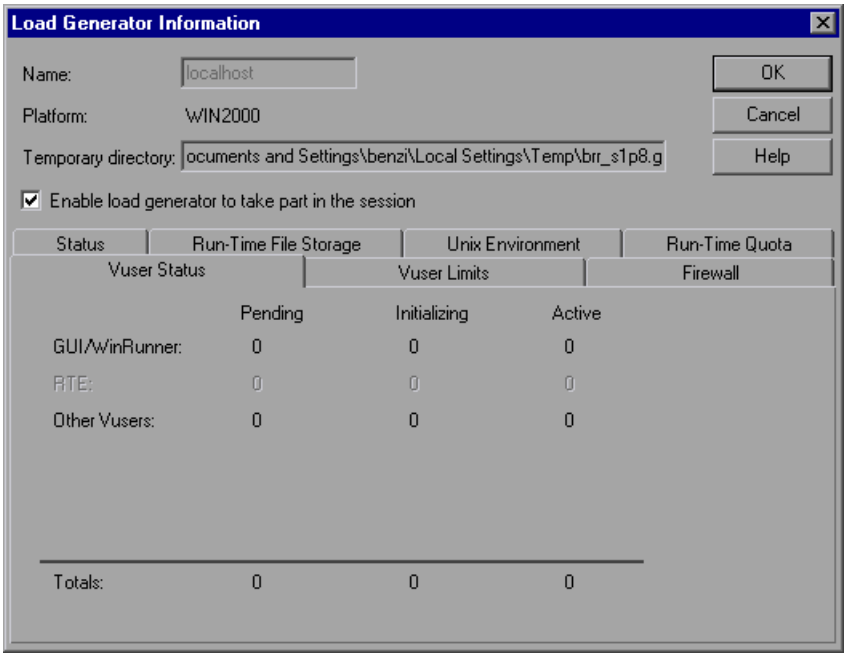
MI Listener: Type the name of the MI listener the load generator is using.

Note: If the load generator is connected, you cannot change values in the **Firewall** tab. To disconnect a load generator, select the load generator in the Load Generators dialog box, and click **Disconnect**. The load generator status changes to DOWN, and you can change the settings.

The Firewall tab is disabled if the load generator is **Localhost**.

Vuser Status Tab

Select the **Vuser Status** tab to view the status of all the Vusers connected to the selected load generator machine.



Note: This tab can only be viewed when the load generator machine is connected.

GUI/WinRunner: Displays the number of GUI/WinRunner Vusers that are *Pending, Initializing*, and *Active*.

RTE: Displays the number of RTE Vusers that are *Pending, Initializing*, and *Active*.

Other Vusers: Displays the number of Vusers—other than GUI/WinRunner and RTE Vusers—that are *Pending, Initializing*, and *Active*.

Totals: Displays the total number of Vusers that are *Pending, Initializing*, and *Active*.

Note: For information on the Connection Log tab available in Expert mode, see "Working in Expert Mode" on page 759.

12

Scheduling Session Steps

After you create a step, you use the Schedule Builder to specify when the step should begin running. In addition, you can set the duration of the step, or specify that a step should be run until it reaches its goal.

This chapter describes:

- Specifying Execution Time
- Creating and Selecting a Profile
- Creating a Manual Profile
- Creating a Goal-Oriented Profile

About Scheduling Session Steps

An important factor in the creation of a test step, is developing a step that accurately portrays user behavior—the types of actions and the timing of those actions, represented by the scripts.

The Schedule Builder allows you to specify when to run steps, and for how long. You do this by creating *profiles* and associating them with the steps. Each profile defines a specific way of testing. By creating multiple profiles you can run the same step under different conditions.

The Schedule Builder allows you to create the following types of profiles: *manual* and *goal-oriented*.

In a manual profile, you specify:

- start time of a test
- duration

➤ number of Vusers that will run the test

➤ the ramp up and ramp down processes

Manual profiles include baseline profiles, which cause LoadRunner Tuning Module to run all the enabled scripts on all the Load Generators.

In a goal-oriented profile, you specify the goal you want to reach (for example, the response time that is considered unacceptable). LoadRunner Tuning Module runs the step, adding Vusers, till the step reaches the goal.

The Schedule Builder's Load Preview graph displays the defined profile.

Specifying Execution Time

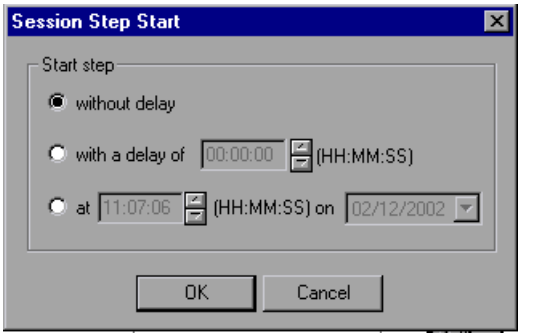
By default, when you issue an *Execute* command, LoadRunner Tuning Module immediately starts the session step. You can instruct LoadRunner Tuning Module to run the step at a later point in time. You do this by specifying one of the following:

➤ The length of time that you want LoadRunner Tuning Module to delay execution after an *Execute* command is issued.

➤ A specific time when you want execution to begin.

To specify when a session step should be executed:

1 Select **Session** > **Start Time**. The Session Step Start dialog box opens, with the default option—**without delay**—selected.



2 Specify when you want the step to start execution.

3 Click **OK** to close the dialog box and save your settings.

The next time you execute a session step, the start time will be delayed as specified.

Understanding the Session Step Start Dialog Box

The Session Step Start dialog box enables you to delay the time at which a session step begins.

Start Step: Select one of the following options:

- **without delay:** Starts the step immediately when you click the Tune button.
- **with a delay of X (HH:MM:SS):** Starts the step after the period you specify has elapsed. Enter the period in hours:minutes:seconds format.
- **at X (HH:MM:SS) on X:** Starts the step on the day and at the time you specify. Specify the time (in hours:minutes:seconds format) and date for starting step execution.

Note: You can also specify the ramp up schedule and duration of a step or Vuser group. See "Understanding the Ramp Up Tab," on page 127 and "Understanding the Step Duration Tab," on page 128.

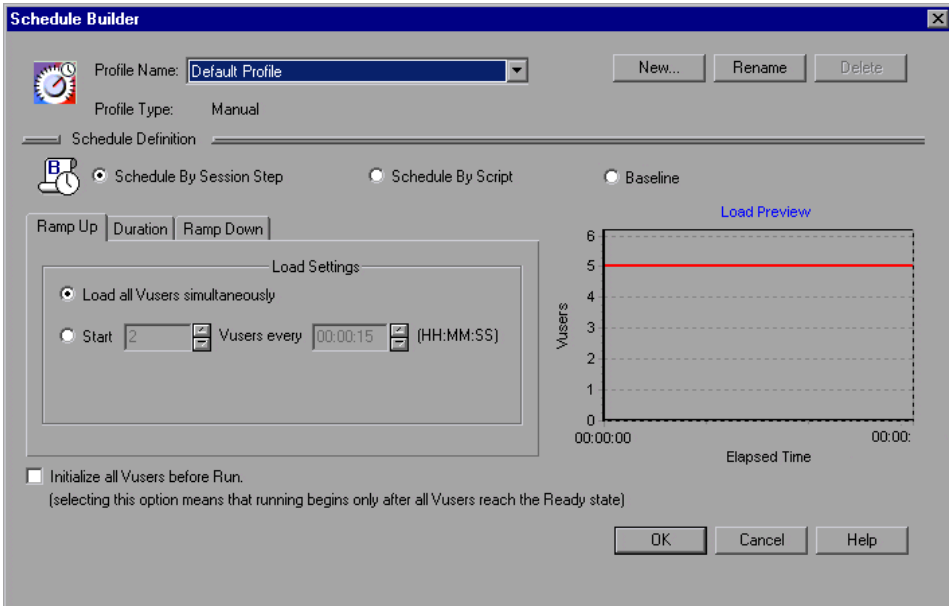
Creating and Selecting a Profile

You use the Schedule Builder for choosing the profile for your step, creating new profiles, and modifying existing ones. You select the profile that you want to use for your session step from the Profile Name list box in the Schedule Builder window. The default profile loads all Vusers simultaneously.

To invoke the Schedule Builder:

- Choose **Session > Schedule Builder** or click one of the Schedule Builder icons (in the toolbar or in the Session tab's Schedule Summary section). The Schedule Builder window opens, displaying the last profile that was associated with the step.

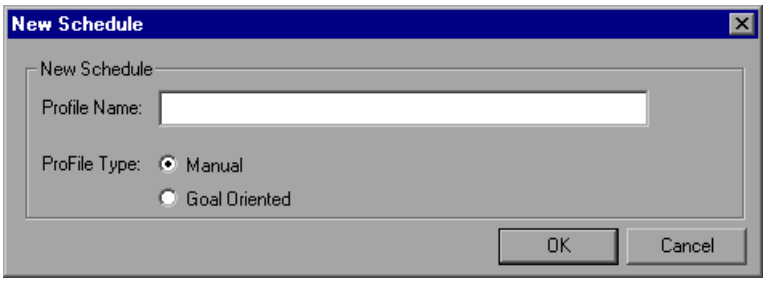
The profile name is displayed in the Profile Name list box. If no profile has been associated with the step, LoadRunner Tuning Module by default uses a manual profile and assigns it the name *Default Profile*.



To create a new profile:

1 Invoke the Schedule Builder.

2 In the Schedule Builder window, click the **New** button. The New Schedule dialog box opens.



3 In the **Profile Name** text box, enter the name of the new profile.

4 Choose the profile type—**Manual** or **Goal-Oriented**—and click **OK**.

The new profile name appears in the Profile Name list box in the Schedule Builder window.

5 Set the values for your profile (see "Creating a Manual Profile," on page 125 or "Creating a Goal-Oriented Profile," on page 132).

To choose an existing profile for your step:

1 Invoke the Schedule Builder.

2 Choose a profile from the Profile Name list.

To modify the properties of an existing profile:

1 Invoke the Schedule Builder.

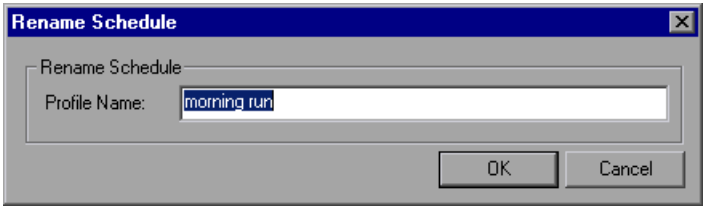
2 Choose the profile you want to modify from the Profile Name list.

3 In the Schedule Builder dialog box, modify the profile as required.

To rename a profile:

1 In the Schedule Builder window, select the profile you want to rename from the Profile Name list.

2 Click **Rename**. The Rename Schedule dialog box appears.



3 Enter a new name for the selected profile, and click OK. The new name appears in the Profile Name list.

To delete a profile:

1 In the Schedule Builder window, select the profile you want to delete from the Profile Name list box.

2 Click **Delete**. The profile is deleted, and no longer appears in the Profile Name list box.

Understanding the Schedule Builder Window

The Schedule Builder window enables you to configure the schedule settings for the step.

Profile Name: Select the name of the profile you want to use for the step. LoadRunner Tuning Module displays the last profile that was associated with the step. The profile name is displayed in the Profile Name list box. If no profile has been associated with the step, LoadRunner Tuning Module by default uses a manual profile and assigns it the name Default Schedule.

Profile Type: Manual or goal-oriented.

New: Opens the New Schedule dialog box, enabling you to enter the new schedule name and specify the Schedule Profile type—manual or goal-oriented.

Rename: Renames the schedule profile.

Delete: Deletes the schedule profile.

Schedule Definition

Defines the settings of the schedule. This section has different settings for manual and goal-oriented schedule profiles.

If the Schedule Builder is displaying a manual profile, LoadRunner Tuning Module displays radio buttons for choosing the scheduling method:

- Schedule By Session Step
- Schedule By Script
- Baseline

Each radio button causes LoadRunner Tuning Module to display a different set of tabs for defining the manual profile.

When you choose **Schedule by Session Step**, LoadRunner Tuning Module displays the following tabs:

- Ramp Up tab (see "Understanding the Ramp Up Tab," on page 127).
- Duration tab (see "Understanding the Step Duration Tab," on page 128).
- Ramp Down tab (see "Understanding the Ramp Down Tab," on page 128).

When you choose **Schedule by Script**, you see the Start Time tab (see "Understanding the Start Time Tab," on page 130), in addition to the tabs you see when you click **Schedule by Session Step**.

When you choose **Baseline**, you see the Define Baseline tab (see "Understanding the Baseline Pane," on page 131).

Goal-Oriented Profile Scheduling Settings

If the current profile is a goal-oriented one, the **Define Step Goal** section is displayed. This is where you define the goal for your step.

In addition, you see the following tabs:

- Step Settings Tab (see "Understanding the Step Settings Tab," on page 136).
- Load Behavior Tab (see "Understanding the Load Behavior Tab," on page 137).

Creating a Manual Profile

A manual profile can use the following types of scheduling:

- Session Step Scheduling
- Script Scheduling
- Baseline Scheduling

Session Step Scheduling

Session step scheduling means creating a profile that specifies the following:

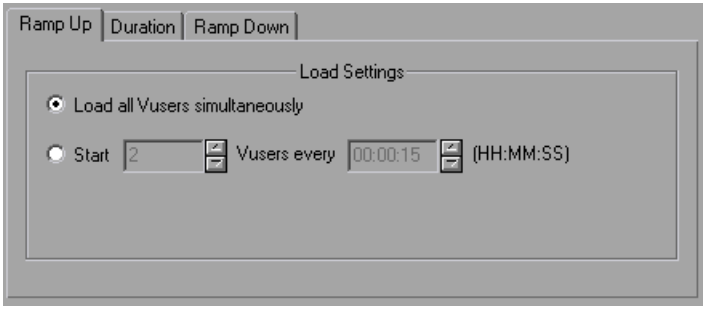
- how the step should be started (ramped up). This allows you to choose between gradually adding Vusers to the running test, or starting all the Vusers simultaneously when the test starts.
- the step duration
- how the step should be stopped (ramped down). This allows you to choose between gradually stopping Vusers that are running, and stopping them all simultaneously.

Note: When you schedule by session step, your settings apply to all the scripts included in the step. For example, if you specify a duration, all the scripts will be executed for the specified period.

To specify how the step should be started, the step duration, and how the step should be stopped:

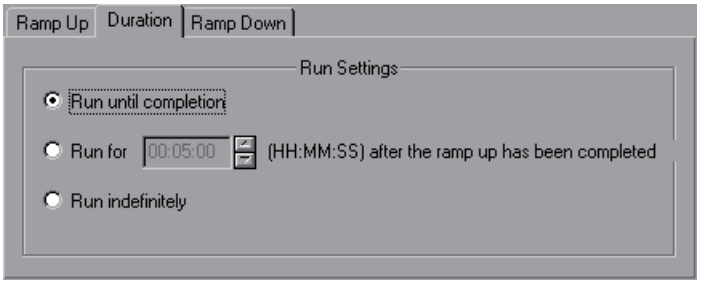
1 In the Schedule Builder window, click **Schedule by Session Step**.

2 Select the **Ramp Up** tab and specify how the step should be started.



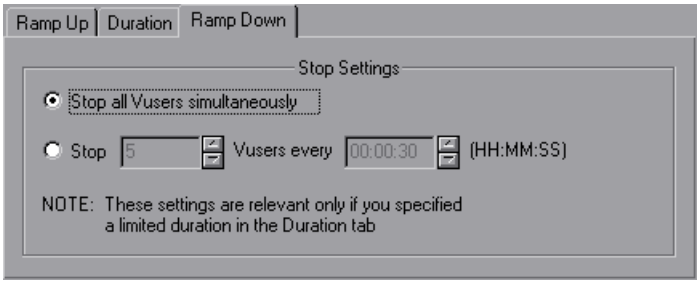
For information on the Ramp Up tab, see "Understanding the Ramp Up Tab," on page 127.

3 Select the **Duration** tab and specify the step's duration.



For information on the Duration tab, see "Understanding the Step Duration Tab," on page 128.

4 Select the **Ramp Down** tab and specify how the Vusers should be stopped.



For information on the Ramp Down tab, see "Understanding the Ramp Down Tab," on page 128.

To instruct LoadRunner Tuning Module to initialize Vusers before beginning to load them:

1 Check the **Initialize all Vusers before run** box. LoadRunner Tuning Module will begin to load the Vusers only after they have all reached the READY state.

2 Click **OK** to close the Schedule Builder and save your settings.

Understanding the Ramp Up Tab

The Ramp Up tab lets you determine how LoadRunner Tuning Module will load the Vusers in a step.

You can select one of the following options:

Load all Vusers simultaneously: Loads and starts all the Vusers in the step at the same time.

Start X Vusers every X (HH:MM:SS): Gradually runs the Vusers. LoadRunner Tuning Module adds a certain number of Vusers to the step within a specified time frame. Enter the number of Vusers you want to begin running concurrently, and the period that you want LoadRunner Tuning Module to wait before adding more Vusers.

Note: While a step is running, you can add scripts to the step and enable them, as long as not all of the Vusers have been ramped up. However, if you add a script after all the Vusers have been ramped up, the new script will not run while the current step is executing. To enable running the new script in the step, you need to stop the step and restart it.

Understanding the Step Duration Tab

Lets you specify the length of time for which a step will run.

You can select one of the following options:

Run until completion: Runs the step once.

Run for X (HH:MM:SS) after the ramp up has been completed: Runs the step for the amount of time you specify, once all the Vusers have been ramped up.

Run indefinitely: Runs the step for an unlimited amount of time.

Note: The duration setting overrides the Vuser iteration settings. This means that if the duration is set to five minutes, the Vusers will continue to run as many iterations as required in five minutes, even if the run-time settings specify only one iteration.

Understanding the Ramp Down Tab

The Ramp Down tab lets you specify how LoadRunner Tuning Module will stop the Vusers in a step.

You can select one of the following options:

Stop all Vusers simultaneously: Stops all the Vusers in the step at the same time.

Stop X Vusers every X (HH:MM:SS): Stops a certain number of Vusers within a specified time frame. For example, you might want to stop 5 Vusers every 30 seconds.

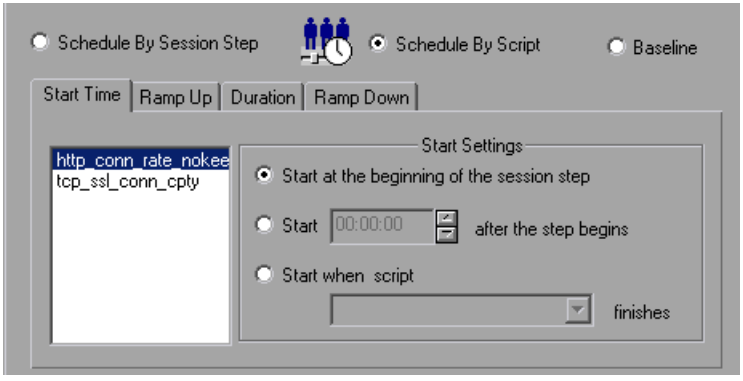
Note: The Ramp Down tab settings are enabled only if you select the second option (**Run for ... after the ramp up has been completed**) in the Duration tab.

Script Scheduling

Script scheduling allows you to specify settings for each script in the step separately. For example, you can schedule a different duration for each script.

To schedule scripts:

1 In the Schedule Builder window, click the Schedule by Script radio button. LoadRunner Tuning Module displays a Start Time tab, in addition to the tabs displayed when you schedule by session step (see Session Step Scheduling on page 125).



2 The Start Time tab includes a list of the scripts contained by the step. To specify settings for a particular script, click the step name to select it and choose the appropriate start time setting.

3 Specify the settings in the other tabs (ramp up, duration and ramp down) as described in "Scheduling Session Steps," on page 119.

Understanding the Start Time Tab

The Start Time tab is used for specifying separate scheduling settings for each script in a step. The tab includes a list of the scripts contained by the step. To specify settings for a particular script, click the script name to select it and specify when you want the script to be executed.

Start at the beginning of the session step: Causes the script to be run at the beginning of the session step.

Start HH:MM:SS after the step begins: Delays running the script for the specified period. You specify the delay period in HH:MM:SS format. LoadRunner Tuning Module will run the script when the specified period has passed after the step has begun executing.

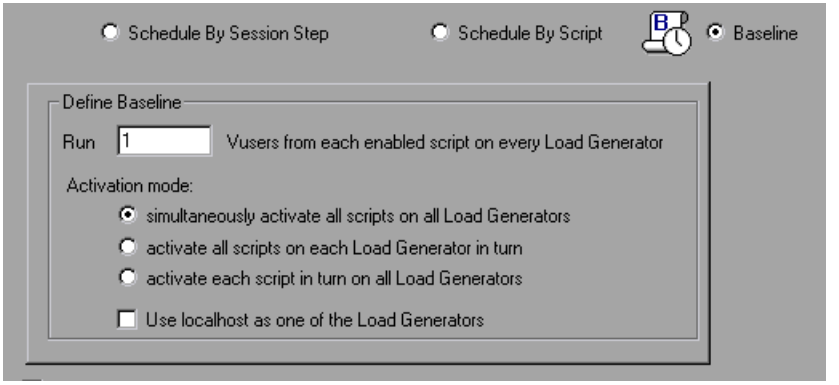
Start when script finishes: Makes the script dependent on another script in the step finishing execution. After clicking the radio button, choose a script from the list box. LoadRunner Tuning Module will run your script only after the specified script has been run.

Baseline Scheduling

Baseline scheduling allows you to test all of the enabled scripts on all the Load Generators (instead of only testing each script on the machine to which it is assigned). Baseline scheduling is particularly useful when dealing with a complicated system topology that has a large number of servers and hosts. Running a baseline test before tuning lets you verify that all the hosts and scripts are valid.

To schedule by baseline:

1 In the Schedule Builder window, click the Schedule by Baseline radio button. LoadRunner Tuning Module displays the Define Baseline pane:



2 Specify the number of Vusers that each load generator or enabled script should run.

3 Specify the activation mode.

Understanding the Baseline Pane

The Baseline pane is used for testing all of the enabled scripts on all the Load Generators (instead of only testing each script on the machine to which it is assigned)

Run ... Vusers: In the text box, enter the number of Vusers that LoadRunner Tuning Module should run from each enabled script or Load Generator. For example, if you enter the number 5, each script will be run by five Vusers.

Simultaneously activate all the enabled scripts on all of the Load Generators: When you choose this mode, LoadRunner Tuning Module activates all of the enabled scripts on all the load generators at the same time.

Activate all the enabled scripts on each Load Generator in turn: When you choose this mode, LoadRunner Tuning Module first runs all the scripts on Load Generator 1, next on Load Generator 2, and so on.

Activate each enabled script in turn on all the Load Generators: When you choose the mode. LoadRunner Tuning Module runs script 1 on Load Generator 1, next on Load Generator 2, and so on until the script has been run on all the Load Generators. LoadRunner Tuning Module next runs script 2 on all of the Load Generators, and so on until all the scripts have been run on all the Load Generators.

Use localhost as one of the Load Generators: Check this box to use the local machine as one of the Load Generators.

Creating a Goal-Oriented Profile

You specify a goal for a session step—not for a script.

When you run a session step, the goal you defined is displayed in the appropriate graph, along with the session results. This enables you to compare the results with your target goal and determine if your goal was reached. If your goal is not reached, you reconfigure your applications and servers accordingly, in order to reach the desired goal.

Creating a goal-oriented profile includes defining the following:

- Step goal—the goal that you want to achieve before terminating execution of the step. This includes the value that you want to achieve (for example, the number of concurrent transactions) and the maximum number of Vusers that will participate in the session step.
- Step settings—when LoadRunner Tuning Module should run the step and what it should do if the goal is not reached. **Note:** Times in the scheduler are approximate; although LoadRunner Tuning Module attempts to reach the target within the specified period, the actual period may be affected by external factors (for example, Internet connections).
- Load behavior—how and when you want LoadRunner Tuning Module to reach your target. LoadRunner Tuning Module starts each step by ramping up.

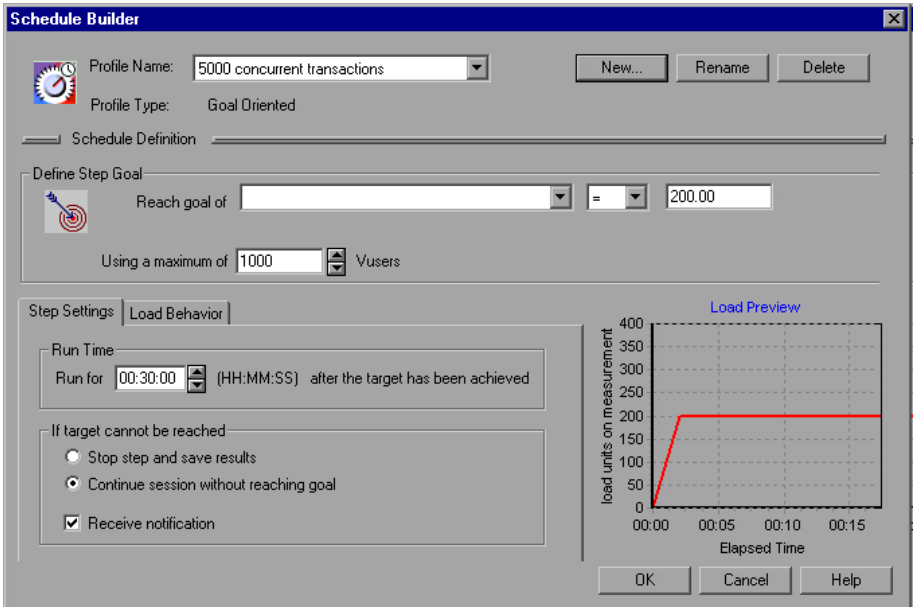
When LoadRunner Tuning Module starts executing a goal-oriented step, it first executes the step for two minutes with one Vuser from each group. Based on the performance that it measures during the two minutes, LoadRunner Tuning Module calculates the number of Vusers needed to reach the goal. The subsequent behavior depends on the selected Ramp Up option (*Automatic* or *Reach target after*) in the Load Behavior tab.

- If you select *Automatic*, the rest of the Vusers needed are run immediately. For example, if LoadRunner Tuning Module was running one Vuser during the initial two minutes, and it calculates that a total of ten Vusers are needed to reach the goal, it starts another nine Vusers and runs them all for another two minutes. If the goal is not reached by the end of this period, LoadRunner Tuning Module calculates the difference between the value that was reached and the goal, and will either start another batch of virtual users or stop some virtual users, depending on whether the value is below the goal or above it, respectively. Note that if you use the equals operator ("=") to specify an exact value for the goal, LoadRunner Tuning Module considers the goal to have been reached only if the actual value stays within 2% of the goal for 60 seconds. For example, if you specify a goal of exactly 100 connections, and the actual number of connections is 110, LoadRunner Tuning Module does not consider the goal to have been reached.
- If you select *Reach target after*, LoadRunner Tuning Module spreads the starting of the Vusers over the time period you specify. For example, if you specify ten minutes and LoadRunner Tuning Module calculates that it needs five Vusers, one Vuser will be started every two minutes. Note that LoadRunner Tuning Module adjusts the number of Vusers that it starts each time (the batch) so that adding each batch results in the same change of measurement over the same time period. If the current batch does not cause the measurement to increase as much as the previous batch, LoadRunner Tuning Module increases the number of Vusers in the batch. Similarly, if the current batch causes a greater change than the previous one, LoadRunner Tuning Module decreases the number of Vusers in the batch.

  Once the goal has been reached, LoadRunner Tuning Module keeps running it for the period specified in the Step Settings section, adding or subtracting Vusers as needed to keep the actual measurement within 6% of the specified target. For example, if the Step Goal is specified as 100 hits/sec, LoadRunner Tuning Module adds or subtracts Vusers as needed to keep between 94 and 106 hits per second.

You can define multiple goal-oriented profiles for each step, and execute the step using a different profile each time.

When you create a goal-oriented profile, the Schedule Builder displays the following window:



To define the step goal:

1 Choose the type of goal for your step from the **Reach goal of** list box.

The list of goal types includes:

➤ basic HTTP-related goals. These goal types are displayed when you open the list box.

➤ additional goals that are related to the monitors and measurements you have specified. To view these goal types, choose **<more>** in the **Reach goal of** box.

Following is the list of basic HTTP-related goals:

- **Throughput**—target downstream bandwidth.
- **Hits / Second**—target number of hits per second (HTTP requests per second) that you would like your step to reach. When you choose this goal, you also need to enter the maximum number of Vusers for the session step.
- **Number of Connections**—target number of connections that you would like the server to host.
- **Connections / Second**—target number of connections per second you would like the server to handle.

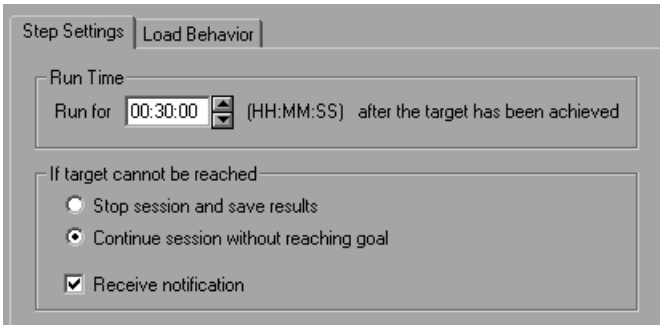
2 Choose an operator (either "=" or ">=")from the middle list box.

3 Enter a value in the text box on the right side.

4 Specify the maximum number of Vusers in the **Using a Maximum of ... Vusers** box.

To define step settings:

1 Click the **Step Settings** tab.



2 Enter a value (in HH:MM:SS format) in the **Run For** box. This value specifies how long the step will run after the goal has been reached.

3 Specify what LoadRunner Tuning Module should do if the goal is not reached: click the appropriate radio button to either stop the session and save the results, or continue the session despite not reaching the goal.

4 Check the Receive Notification box if you want LoadRunner Tuning Module to display a message when it determines that the goal cannot be reached.

Understanding the Step Settings Tab

The Step Settings Tab lets you specify how long you want your step to run after the target is achieved, and whether to continue the step if the target you defined cannot be reached.

Note: Times in the scheduler are approximate; although LoadRunner Tuning Module attempts to reach the target within the specified period, the actual period may be affected by external factors (for example, Internet connections).

Run Time

➤ **Run for X (HH:MM:SS) after the target has been achieved:** Select the amount of time you want your step to run after reaching the target.

If target cannot be reached: Select one of the following two options:

➤ **Stop session and save results:** Instructs the Console to stop the session and save the session results, if the target you defined cannot be reached.

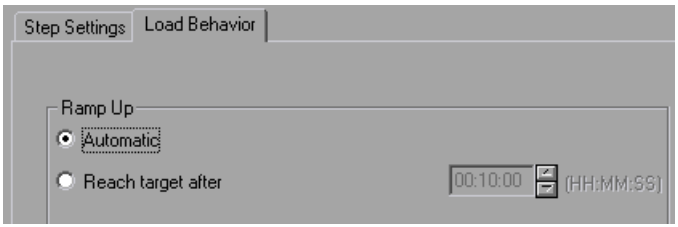
➤ **Continue session without reaching goal:** Instructs the Console to continue running the session, even if the target you defined cannot be reached.

Receive notification: Instructs the Console to send you an error message indicating that the target cannot be reached.

Defining Load Behavior

To define load behavior:

1 Click the **Load Behavior** tab.



2 Specify the ramp up.

Understanding the Load Behavior Tab

The Load Behavior tab lets you specify how and when you want the Console to reach your target.

LoadRunner Tuning Module starts each step by ramping up. LoadRunner Tuning Module starts the step's ramp up stage by executing a batch of Vusers, one Vuser for each script defined in the step. After two minutes of step execution, LoadRunner Tuning Module calculates the number of Vusers needed to reach the goal. The subsequent behavior depends on the selected Ramp Up option.

Ramp Up: Select one of the following options:

- **Automatic:** Start all the required Vusers simultaneously.
- **Reach target after:** Specify the total time required for starting all the batches of Vusers. Enter the period in the box in HH:MM:SS format. (Minimum: 2 minutes.) LoadRunner Tuning Module starts a batch of Vusers approximately every 2 minutes, and attempts to make the batches equal in size.

Note: Reaching the goal depends on the maximum number of Vusers specified in the Define Step Goal settings (see "Goal-Oriented Profile Scheduling Settings," on page 125.

13

Defining Alerts

Before tuning your session, you set up alerts that define what actions LoadRunner Tuning Module should take when server performance problems occur.

This chapter describes:

- Types of Alerts
- Understanding the Alerts Window
- Alert Conditions
- Alert Actions
- Creating, Configuring, and Deleting Alerts
- Enabling and Disabling the Alert Mechanism
- Viewing Alerts in the Output Window

About Defining Alert Schemes

LoadRunner Tuning Module uses alerts to let you know when server performance problems occur.

Before running scripts, you define alerts for one or more measurements. This includes specifying what conditions should trigger an alert, and the action that LoadRunner Tuning Module should take when the need to issue an alert is detected.

You use the Alerts window to specify alert conditions and an alert scheme. You specify separate alert conditions for each session step.

Note: Although you select measurements to monitor topology elements (not physical components), you assign alerts to physical machines. This means that if an alert is triggered on a measurement of one topology element, it affects all the other elements mapped to the same physical machine. For example, if the alert action is to stop Vusers or stop the ramp-up, the action will affect all the Vusers, regardless of the load generators on which they run and the machines against which they are running.

Types of Alerts

You can create alerts that will be triggered by specific values occurring in the following measurements:

Note: The following is a partial list of the measurements that you can use for triggering alerts. All measurements that are available in the Console can be used.

➤ **Running Vusers**

Informs you when the number of Running Vusers in the Running, Ready, Finished, or Error state, reaches a specific value.

➤ **Error Statistics**

Informs you when the number of errors reaches a specified number.

➤ **Vusers with Errors**

Informs you when the number of Vusers with errors reaches a specified number.

➤ **Transaction Response Time (Passed)**

Informs you when the transaction response time reaches a specified value, for the *vuser_init*, *Actions*, or *vuser_end* sections of the script.

➤ **Transactions Per Second (Passed)**

Informs you when the specified number of transactions per second is reached for the *vuser_init*, *Actions*, or *vuser_end* sections of the script.

➤ **Total Transactions Per Second (Passed)**

Informs you when the specified number of transactions per second is reached for the *vuser_init*, *Actions*, or *vuser_end* sections of the script.

➤ **Hits per Second**

Informs you when the specified number of hits per second is reached.

➤ **Throughput**

Informs you when your server's throughput (downstream bandwidth) reaches a specific level.

➤ **Pages Downloaded per Second (Passed)**

Informs you when a specific number of pages has been downloaded per second by the server.

Alert Conditions

LoadRunner Tuning Module allows you to specify when you want it to issue alerts for the measurement that you are monitoring. LoadRunner Tuning Module provides several types of alert schemes. The alert schemes instruct

LoadRunner Tuning Module to issue an alert when it detects a specific value, a value for a specific duration, an out of range value, a change in value, or a standardized value.

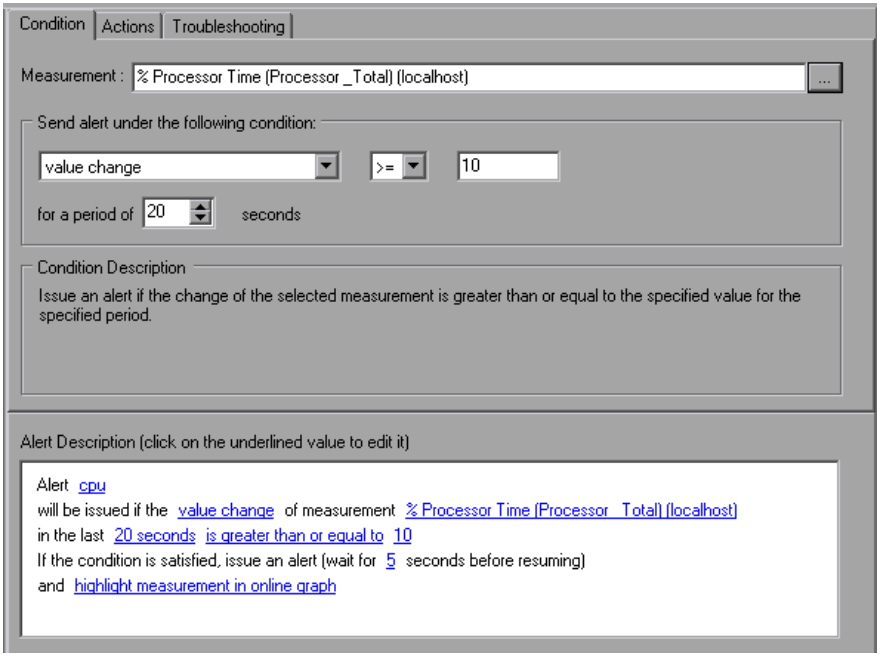
Scheme	Additional fields	Meaning
value	none	Issue an alert if the measurement compares with the specified value in one of the following ways: > >= < <= =
value for a duration of time	for a period of *n* seconds	Issue an alert if the selected measurement compares with the specified value in one of the following ways for *n* seconds: > >= < <= =
value out of range	range	Issue an alert if the measurement deviates from the specified range.

Scheme	Additional fields	Meaning
value change	for a period of *n* seconds	Issue an alert if the change in the selected measurement compares with the specified value in one of the following ways over the last *n* seconds: > >= < <= =
standardized value	for a period of *n* seconds	Issue an alert if the standardized value of the selected measurement compares with the specified value in one of the following ways, considering the period of the last *n* seconds : > >= < <= =

You can choose one of the following conditions for your alert scheme:

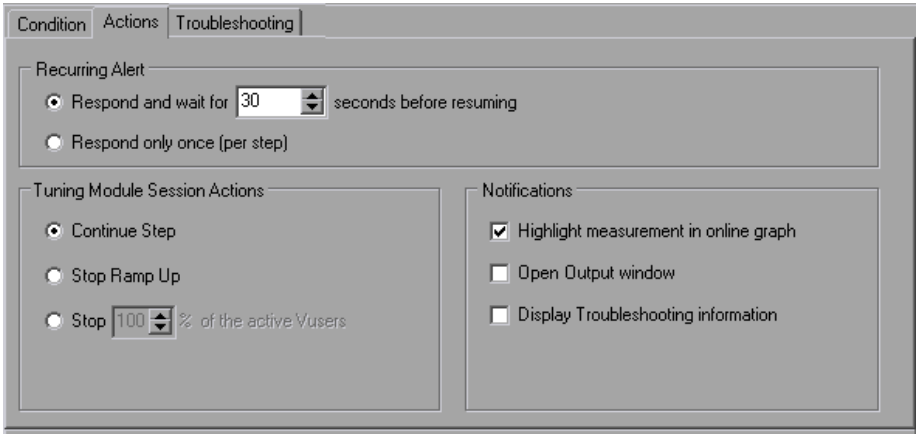
- > greater than
- >= greater than or equal to
- < less than
- <= less than or equal to
- = equal to

You define an alert by selecting a measurement, scheme, and condition in the Alerts Window's Condition tab. In the following example, LoadRunner Tuning Module is instructed to issue an alert when it detects a change of 10 or more in the value of the *% Processor Time* measurement, over a period of 20 seconds.



Alert Actions

The Alerts Window's Actions tab allows you to specify the actions LoadRunner Tuning Module takes when an alert is triggered.



LoadRunner Tuning Module also assigns default settings for the actions that are triggered by an alert. LoadRunner Tuning Module displays these settings in the Actions tab. You can change the default settings to suit your needs.

You can specify settings for the following:

- Recurring Alerts
- Session Actions
- Notification Types

Understanding the Actions Tab

You use the Actions tab to configure the way LoadRunner Tuning Module sends alert notifications.

The Actions tab lets you specify the following:

Recurring Alert: Relates to the frequency of responses for recurring alerts in a single step. Select one of the following options:

- **Respond and Wait *n* seconds before resuming**—specify a value for *n*.

➤ **Respond only once (per step)**

The default is to respond and wait 30 seconds before resuming the test.

Tuning Module Session Actions: Specify what action to perform when encountering an alert condition. Select one of the following options:

➤ **Continue Step:** Continues executing the current step, even after an alert trigger.

➤ **Stop Ramp Up:** Stop adding additional Vusers to the session step.

➤ **Stop *n* % of the active Vusers:** Stops a percentage of the Vusers when an alert is triggered. Specify a value for *n* between 0 and 100. (Default: 100).

Notification: Specify the notifications to issue when an alert is triggered. The available notification types are:

➤ **Highlight measurement on online graph:** Highlights the measurement that triggered the alert on the online graph.

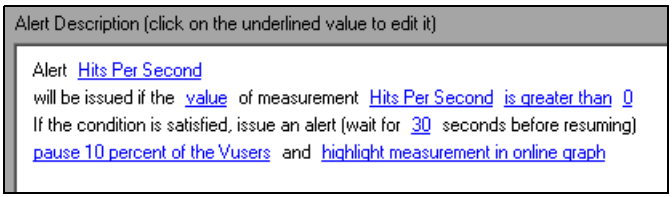
➤ **Open Output window:** Opens the Output window to display alert messages when an alert trigger occurs. For more information about viewing alert messages in the Output Window, see "Viewing Alerts in the Output Window," on page 155.

➤ **Display Troubleshooting information:** Displays troubleshooting information when an alert trigger occurs. See "Alert Troubleshooting Information," on page 153 for details.

You can enable none or all of these notification mechanisms. If you do not enable a notification mechanism, you can still view the alerts that occurred in the Output window.

Alert Description: Displays all of the configuration parameters of the Alerts window in plain language. It includes hyperlinks that help you to change the alert's condition and actions. To modify any of the values or settings,

click on an underlined value. LoadRunner Tuning Module places the cursor in the relevant field within the Alerts window.



In some cases, clicking a hyperlink causes LoadRunner Tuning Module to open a dialog box so you can change the relevant value.

Options: Click to open the Alert Options dialog box where you can set the alert triggers or enable/disable alerts. The Alerts Options dialog box opens. Select a trigger option and click **OK**. For more information, see the section "Enabling and Disabling the Alert Mechanism," on page 154.

Creating, Configuring, and Deleting Alerts

After designing the session steps, you use the Alerts Window to set alerts for your tuning session. The alerts provide real-time information about your server performance and inform you when specific thresholds are reached.

After creating an alert, you specify the alert conditions and actions.

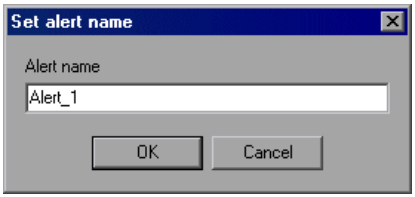
Creating an Alert

Creating an alert includes specifying the alert name and choosing the measurement that will trigger the alert.

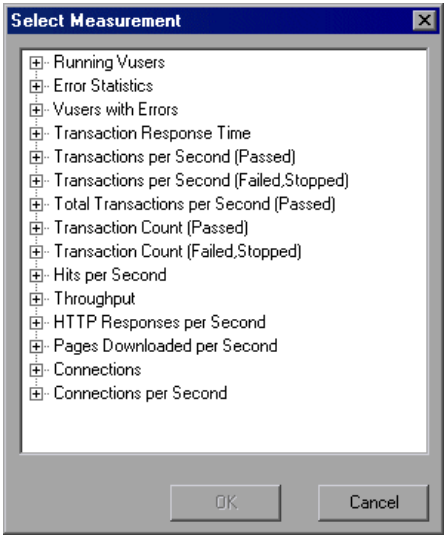
To create an alert:

1 Click the **Alerts Definition** button in the LoadRunner Tuning Module Console window. The Alerts window opens.

2 Click **Add** in the bottom left corner. The Set Alert Name dialog box opens.



3 Enter a name for the alert and click **OK**. The Select Measurement dialog box opens, displaying categories of measurements that you can monitor.



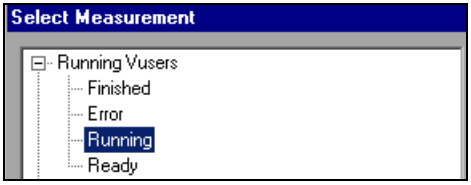
The following categories always appear, regardless of your topology and monitors:

- Running Vusers
- Total Transactions per Second (Passed)
- Hits per Second
- Throughput
- Pages Downloaded per Second
- Connections

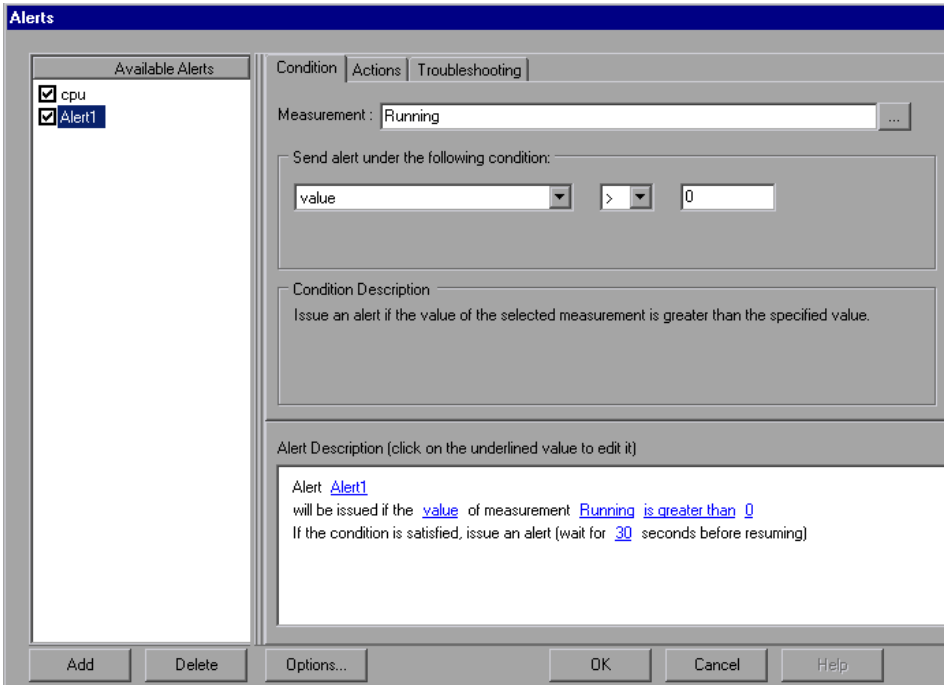
➤ Connections per second

Additional categories may appear, depending on your topology and the monitors you assigned to various elements.

4 Expand the required category to display its measurements.



5 Select the desired measurement and click **OK**. LoadRunner Tuning Module creates an alert with default settings. The alert name you supplied appears in the Available Alerts pane, the Condition and Actions tabs show the default values, and the Alert Description pane displays a plain language description of the alert.



6 To change the default alert condition and actions, click the Condition or Actions tab, respectively, and modify the default settings. For details, see "Understanding the Condition Tab," on page 151 and "Understanding the Actions Tab," on page 145.

7 To specify general alert options, click **Options**.

8 To delete an alert, select it in the Available Alerts pane and click **Delete**.

Understanding the Alerts Window

The Alerts window enables you to add and remove alert conditions for your step. You define certain thresholds for each of the resources that you are measuring. If the threshold is reached, LoadRunner Tuning Module issues an alert.

Available Alerts: Displays a list of all the enabled and disabled alerts in the step.

Add: Type the name of a new alert you want to add to the Available Alerts list.

Delete: Deletes an alert from the Available Alerts list.

Options: Opens the **Alert Options** dialog box, where you can enable, disable or limit LoadRunner Tuning Module alerts. For more details, see "Enabling and Disabling the Alert Mechanism," on page 154.

Condition Tab—For specifying the alert conditions.

Action Tab—For specifying the alert action.

Troubleshooting tab—Information that can be displayed to the user when an alert is triggered.

Alert Actions for Multiple Elements on the Same Host

As mentioned above, an alert that is triggered by a measurement on one topology element also affects other elements on the same physical host, so if the alert action is stopping Vusers or stopping ramp-up, all the Vusers on the physical machine are affected.

However, if you specify the action **Highlight measurement in online graph**, the Console indicates the affected elements as follows when the alert is triggered:

- If the measurement that triggered the alert exists in all the topology elements on the host, the alert causes all the elements to blink in the topology window and the relevant graphs. For example, if the alert was defined on the CPU Utilization measurement, all the elements on the host will blink, along with the CPU Utilization graphs for each element.
- If the measurement that triggered the alert exists in only one of the elements, only that element (and its graph) will blink when the alert is triggered. For example, if the alert was defined for an Oracle measurement, only the element representing the Oracle database will blink.

Understanding the Condition Tab

You use the Condition tab to select a measurement, scheme, and condition for an alert. An alert is sent if the selected conditions are satisfied.

Measurement: Select a measurement to use in your alert. Use the Browse button to display the available measurements. See "Types of Alerts," on page 140.

Send alert under the following condition: Select a scheme and a condition for the measurement.

- Choose one of the following operators for the condition: >, >=, <, <=, or =.
- For the **value** scheme, specify a trigger value.
- For the **value for a duration of time** scheme, specify a trigger value and a duration.
- For the **value out of range** scheme, specify start and end range trigger values.
- For the **value change** scheme, specify a trigger value and a watch time—the alert is only triggered if the condition is reached within the watch time.

Note: If you want LoadRunner Tuning Module to issue an alert when the value decreases by more than 5, specify the "<" operator and a trigger value of -5.

➤ For the **standardized value** scheme, specify a trigger value and a watch time—the alert is only triggered if the condition is reached within the watch time.

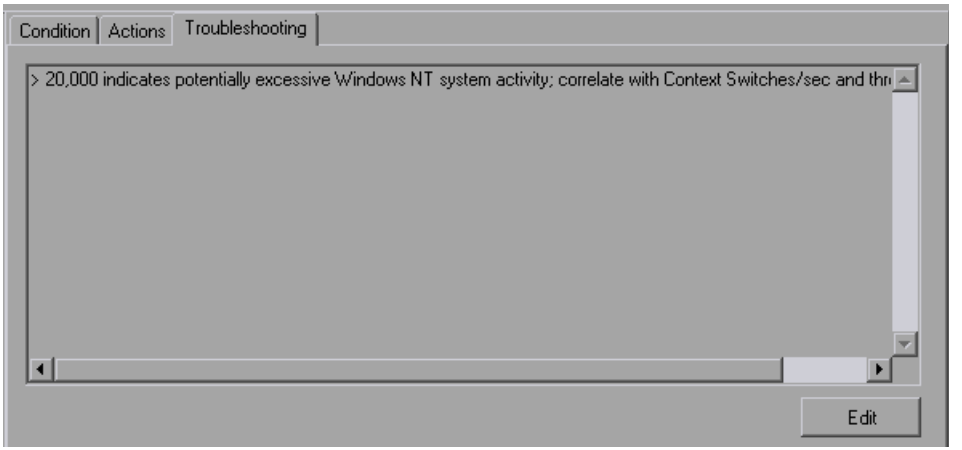
Condition Description: Displays the description of the condition that will trigger an alert.

Note: If you want LoadRunner Tuning Module to issue an alert when a value decreases by more than 5, specify the "<" operator and a trigger value of -5.

Alert Description: Displays all of the configuration parameters of the Alerts window. To modify any of the values or settings, click on an underlined value. LoadRunner Tuning Module places the cursor in the relevant field within the Alerts window.

Alert Troubleshooting Information

The Alerts Window's Troubleshooting tab displays information and recommendations on how the user should react when a specific alert is triggered.



To cause LoadRunner Tuning Module to display the troubleshooting information when the alert is triggered:

➤ In the Actions tab, check the **Display Troubleshooting information** box.

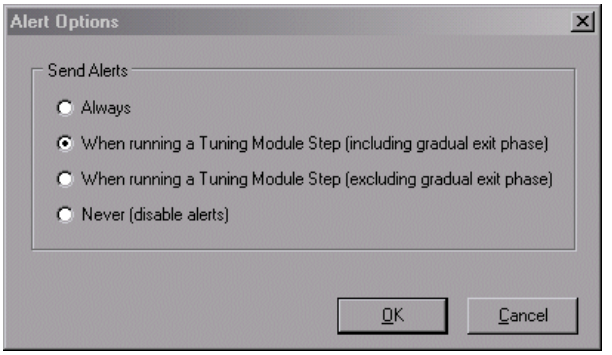
You can add to or change the information that is displayed.

To add to or change the troubleshooting text:

➤ Click **Edit**, change the text, and click **Edit** again.

Enabling and Disabling the Alert Mechanism

After creating alerts, you can specify when (and whether) you want LoadRunner Tuning Module to generate them. In the Alerts Window, click **Options**. The Alert Options dialog box is displayed.



Understanding the Alert Options Dialog Box

You use the Alert Options dialog box to enable, disable, or limit alerts.

Send Alerts: Select one of the following options:

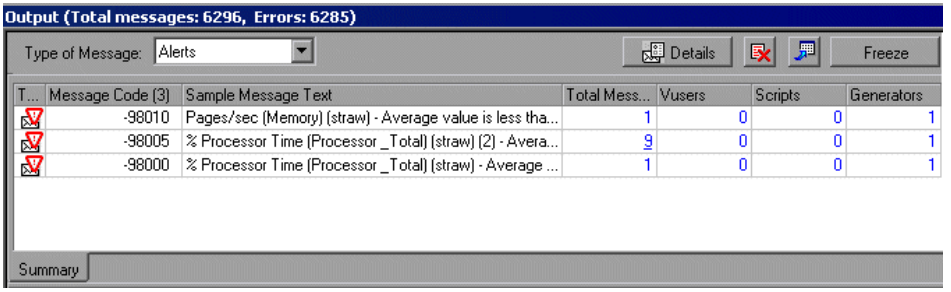
- **Always:** Always issue alerts when the alert condition is met, even when LoadRunner Tuning Module is not executing a step.
- **When Running a LoadRunner Tuning Module step (including gradual exit):** Issue alerts only during test execution, even during the Vuser's gradual exit stage.
- **When Running a LoadRunner Tuning Module step (excluding gradual exit phase):** Issue alerts only during test execution, except during the Vuser's gradual exit stage. This is the default option.
- **Never:** Never issue alerts. This disables the alert mechanism for the active session.

Viewing Alerts in the Output Window

The Output window lets you view information about all the alerts triggered during a tuning session.

To open the Output window:

1 Select **View** > **Show Output**. The Output window is displayed.



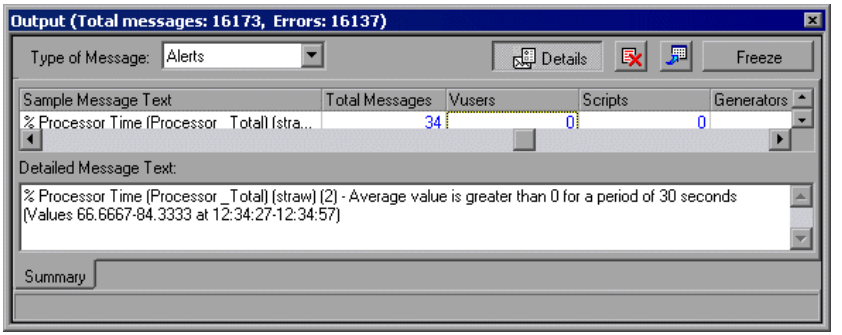
2 If the Type of Message field does not show the word “Alerts”, choose it from the drop-down list.

The Output window now displays a list of all the alerts that have occurred during the session.

The Alerts Output row heading in the title bar shows the total number of messages that have been generated during the session, and the number of errors.

Note that alert information is only saved if alerts were enabled during the script’s execution. If you enabled alerts after script execution, you need to run the step again in order to generate alerts.

3 To show alert details, select an alerts and click **Details**. The Detailed Message Text section opens in the lower part of the Output window.



For more details on using the Output window, see "Viewing the Output Window," on page 212.

14

Configuring Session Steps

You can configure how load generators and Vusers behave when you run a session, so that the session accurately emulates your working environment.

This chapter describes:

- Configuring Session Run-Time Settings
- Setting Timeout Intervals
- Setting the Run-Time File Location
- Specifying Path Translation

About Configuring a Session

Before you run a session, you can configure both the load generator and Vuser behaviors for the session. Although the default settings correspond to most environments, LoadRunner Tuning Module allows you to modify the settings in order to customize the session behavior. The settings apply to all future session runs and generally only need to be set once.

The settings described in this chapter apply to all the load generators in a session. To change the settings for individual load generator machines, refer to Chapter 5, "Creating a Topology." If the global session settings differ from those of an individual load generator, the load generator settings override them.

The settings discussed in this chapter are unrelated to the Vuser run-time settings. These settings, which apply to individual Vusers or scripts, contain information about logging, think time, and the network, the number of iterations, and the browser. For information on setting the run-time settings, see the *Creating Vuser Scripts* guide.

For information on setting the options for online monitors, see Chapter 25, "Online Monitoring."

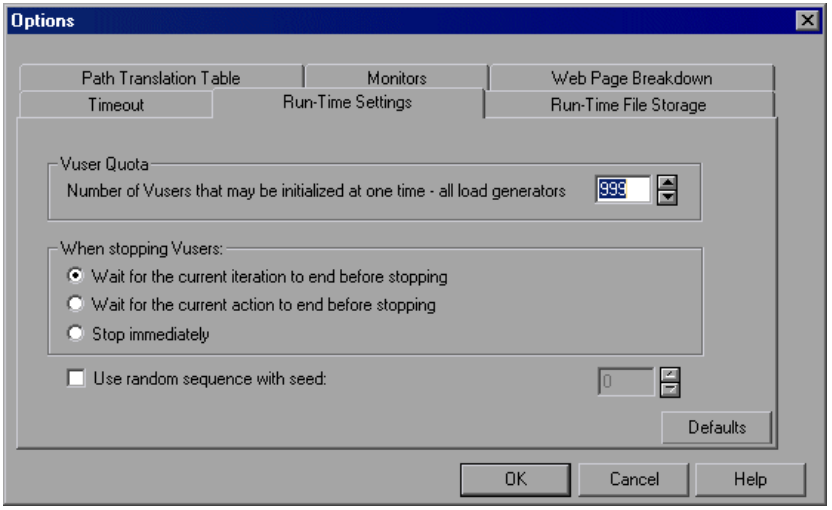
The LoadRunner Tuning Module Expert mode allows you to configure additional settings for the LoadRunner Tuning Module agent and other LoadRunner Tuning Module components. For more information, see Appendix C, "Working in Expert Mode."

Configuring Session Run-Time Settings

You can specify session run-time settings relating to Vuser quotas, stopping Vusers, and random sequence seed using the Run-Time Settings tab.

To set the session run-time settings:

1 Choose **Tools** > **Options**. The Options dialog box opens. Click the **Run-Time Settings** tab.



2 To set a Vuser quota, specify the desired value.

3 Select the way in which you want LoadRunner Tuning Module to stop running Vusers.

4 To specify a seed value for a random sequence, select the **Use random sequence with seed** check box and enter the desired seed value.

Understanding the Options - Run-Time Settings Tab

The Run-Time Setting Tab lets you specify run-time setting values such as the Vuser quota, the way in which Vusers are stopped, or the seed for random sequences.

Vuser Quota: To prevent your system from overloading, you can set quotas for Vuser activity. The Vuser quotas apply to Vusers on all load generators.

- **Number of Vusers that may be initialized at one time - all load generators:** Sets the maximum number of Vusers that the load generator can initialize at a time (when you send an Initialize command).

When stopping Vusers: Lets you control the way in which Vusers stop running when you click the Stop button.

Select one of the following options:

- **Wait for the current iteration to end before exiting:** Instructs LoadRunner Tuning Module to allow a Vuser to complete the iteration it is running before stopping. The Vuser(s) move to the GRADUAL EXITING status and gradually stop executing.
- **Wait for the current action to end before exiting:** Instructs LoadRunner Tuning Module to allow a Vuser to complete the action it is running before stopping. The Vuser(s) move to the GRADUAL EXITING status and gradually stop executing.
- **Stop immediately:** Instructs LoadRunner Tuning Module to stop running the Vuser(s) immediately. The Vuser(s) move to the EXITING status and immediately stop executing.

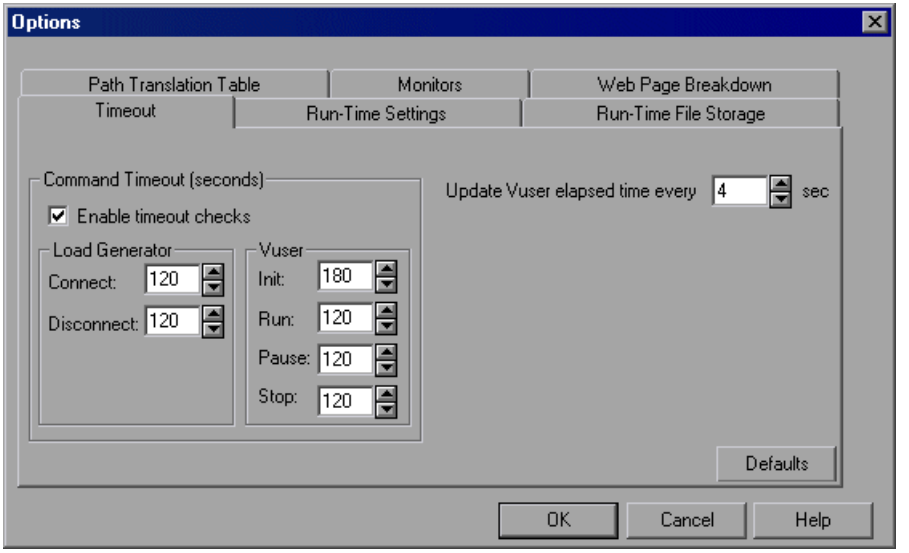
Use random sequence with seed: Allows LoadRunner Tuning Module to use a seed number for random sequencing. Each seed value represents one sequence of random values used for test execution. Whenever you use this seed value, the same sequence of values is assigned to the Vusers in the step. This setting applies to parameterized Vuser scripts using the Random method for assigning values from a data file. It will also affect the random percentage of recorded think time (see information on the Run-Time Settings dialog box in the VuGen Help project). Enable this option if you discover a problem in the test execution and want to repeat the test using the same sequence of random values.

Setting Timeout Intervals

The Timeout Tab lets you specify timeout values for certain commands related to the load generator. If the command is not executed successfully within the timeout period, the load generator status changes to ERROR.

To set timeout intervals:

1 Choose **Tools** > **Options**. The Options dialog box opens. Click the **Timeout** tab.



2 To specify a command timeout interval, select the **Enable timeout checks** check box and specify the appropriate timeouts. Clear the **Enable timeout checks** check box to disable the timeout test.

3 Specify the frequency at which LoadRunner Tuning Module updates the Elapsed time, in the **Update Vuser elapsed time every** box.

Understanding the Options - Timeout Tab

LoadRunner Tuning Module enables you to set the timeout interval for commands and Vuser elapsed time.

The command timeouts are the maximum time limits for various LoadRunner Tuning Module commands. When a command is issued by the

Console, you set a maximum time for the load generator or Vuser to execute the command. If it does not complete the command within the timeout interval, the Console issues an error message.

Command Timeout (seconds)

Enable timeout checks: Instructs LoadRunner Tuning Module to monitor the status of load generators and Vusers after a command is issued by the Console. If the load generator or Vuser does not complete the command within the timeout interval you specified, the Console issues an error message. If you disable the timeout limitations, LoadRunner Tuning Module waits an unlimited time for the load generators to connect and disconnect, and for the Initialize, Run, Pause, and Stop commands to be executed.

➤ **Load Generator**

- ➤ **Connect:** Enter the time limit that LoadRunner Tuning Module waits to connect to any load generator. If a connection is not successful within this time, the status of the load generator changes to FAILED. The default connection timeout is 120 seconds.
- ➤ **Disconnect:** Enter the time limit that LoadRunner Tuning Module waits to disconnect from any load generator. If a disconnection is not successful within this time, the status of the load generator changes to FAILED. The default disconnection timeout is 120 seconds.

Note: LoadRunner Tuning Module recognizes the fact that the number of active Vusers influences the timeout values. For example, 1000 Vusers trying to initialize will take much longer than 10 Vusers. LoadRunner Tuning Module adds an internal value, based on the number of active Vusers, to the specified timeout value.

➤ **Vuser**

- ➤ **Init:** Enter the timeout value for the Initialize command. The default time limit is 180 seconds.
- ➤ **Run:** Enter the timeout value for the Run command. The default time limit is 120 seconds.

➤ **Pause:** Enter the timeout value for the Pause command. The default time limit is 120 seconds.

➤ **Stop:** Enter the timeout value for the Stop command. The default time limit is 120 seconds.

Update Vuser elapsed time every: Specifies the frequency at which LoadRunner Tuning Module updates the value displayed in the Elapsed Time column in the Vusers dialog box. The default is 4 seconds.

Example:

If you select a Vuser and click the **Initialize** button, LoadRunner Tuning Module checks whether the Vuser reaches the READY state within 180 seconds (the default **Init** timeout period); if it does not, the Console issues a message indicating that the **Init** command timed out.

Setting the Run-Time File Location

When you run a session, by default the run-time files are stored locally on each Vuser load generator (the machine running the script). The default location of the files is under the temporary directory specified by the load generator's environment variables (on Windows, TEMP or TMP and on UNIX, $TMPDIR or $TMP). If no environment variable is defined, the files are saved to the /tmp directory.

Note: The run-time file storage settings that are described in this apply to all the load generators in a session. You can change the settings for individual load generator machines as described in "Configuring Load Generators" on page 101.

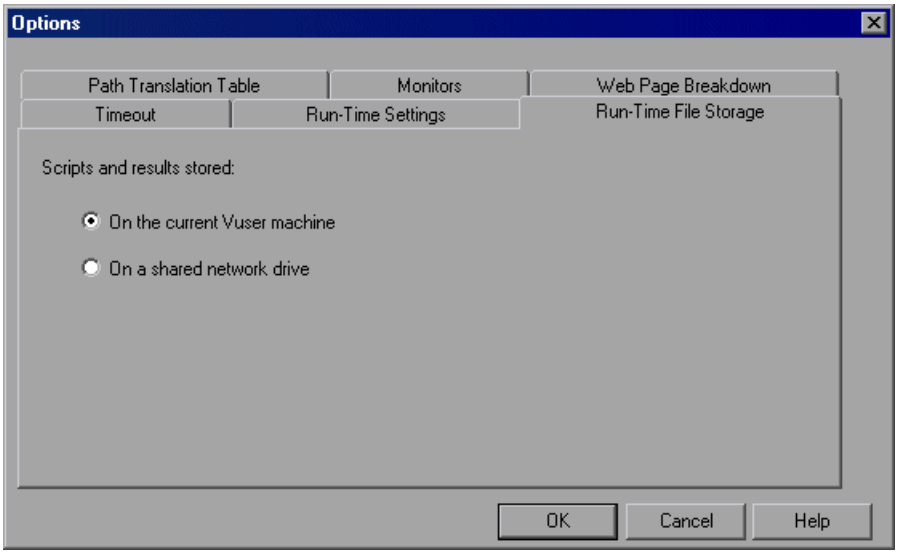
The primary run-time files are script and result files:

Script files: When you run a Vuser, the Console sends a copy of the associated script to the Vuser load generator. The script is stored in the load generator's temporary run-time directory.

Result files: While you run a session, the participating Vusers write their results to the temporary run-time file directory. After session execution, these result files are collated or consolidated—results from all of the load generators are transferred to the results directory. You set the location of the results directory as described in Chapter 17, "Running an Execution Step." After collating the results, the temporary run-time directory is deleted.

To specify where LoadRunner Tuning Module stores run-time files:

1 Choose **Tools** > **Options**. The Options dialog box opens. Click the **Run-Time File Storage** tab.



By default, the **On the current Vuser machine** option is selected. This means that all run-time files—including result files and script files—are stored on the Vuser load generators. The only exception is for Vusers running on the local load generator (Console machine), where you must use the shared drive option.

2 To store script and result files on a shared network drive, click **On a shared network drive**. To set the exact location on the network drive, see Chapter 16, "Preparing to Run a Session Step."

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

Understanding the Options - Run-Time File Storage Tab

The Run-Time File Storage tab lets you specify where LoadRunner Tuning Module saves **Run-Time Files**.

Scripts and results stored: Select one of the following options:

➤ **On the current Vuser machine:** Instructs the Console to save the run-time files on the computer that is running the Vuser script. On an NT-based computer, the results are saved to the directory defined by the TEMP or TMP environment variables. On a UNIX machine, the results are saved to the directory defined by the TMPDIR environment variable. If the TMPDIR environment variable is not defined, the results are saved to the /tmp directory.

Note: If you choose to save result files on the Vuser load generators, you must collate the results before you can perform any analysis. You can wait for LoadRunner Tuning Module to collate the results when you launch the Analysis tool, or you can collate results by selecting **Results** > **Collate Results**. Alternatively, select **Results** > **Auto Collate Results** to automatically collate the results at the end of each session run.

➤ **On a shared network drive:** Instructs the Console to save the step results and/or the Vuser scripts on a shared network drive. A shared network drive is a drive to which the Console and all the load generators in the step have read and write permission. If you select to save results to a shared network drive, you may need to perform path translation. Path translation ensures that the specified results directory is recognized by the remote load generator. For information about path translation see Appendix D, "Performing Path Translation."

If you specify that all Vusers access their Vuser scripts directly at some shared location, no transfer of script files occurs at run time. This alternative method may be useful in either of the following situations:

➤ The file transfer facility does not work.

➤ The Vuser script files are large and therefore take a long time to transfer. Remember that Vuser script files are transferred only once during a session.

This alternate method often necessitates path translation. For details, see Appendix D, "Performing Path Translation."

Specifying Path Translation

If you specified a shared network drive for run-time file storage, (see "Setting the Run-Time File Location" on page 163), you may need to perform *path translation*. Path translation is a mechanism used by LoadRunner Tuning Module to convert a remote path names. A typical session may contain several load generator machines that map the shared network drive differently. For more information, see Appendix D, "Performing Path Translation."

15

Using Rendezvous Points

LoadRunner Tuning Module allows you to check your system's response under specific load. To do this, you can use **rendezvous points** to cause multiple Vusers to perform tasks at exactly the same time, thereby creating intense user load on the server.

This chapter describes:

- Setting the Rendezvous Attributes
- Setting the Rendezvous Policy
- Disabling and Enabling Rendezvous Points
- Disabling and Enabling Vusers at Rendezvous Points
- Viewing Rendezvous Information

About Using Rendezvous Points

During a session step run you can instruct multiple Vusers to perform tasks simultaneously by using rendezvous points. A rendezvous point creates intense user load on the server and enables LoadRunner Tuning Module to measure server performance under load.

Suppose you want to measure how a web-based banking system performs when ten Vusers simultaneously check account information. In order to emulate the required user load on the server, you instruct all the Vusers to check account information at exactly the same time.

You ensure that multiple Vusers act simultaneously by creating a **rendezvous point**. When a Vuser arrives at a rendezvous point, it is held there by the Console. The Console releases the Vusers from the rendezvous either when the required number of Vusers arrives, or when a specified amount of time has passed. For details on the release criteria, see "Setting the Rendezvous Policy," on page 171.

You define rendezvous points in the Vuser script. For information about inserting rendezvous points into Vuser scripts, refer to the *Creating Vuser Scripts* guide.

Using the Console, you can influence the level of server load by selecting:

- which of the rendezvous points will be active during the session step
- how many Vusers will take part in each rendezvous

For example, to test a bank server, you could create a session step that contains two rendezvous points. The first rendezvous ensures that one thousand Vusers simultaneously deposit cash. The second rendezvous ensures that another thousand Vusers simultaneously withdraw cash. If you want to measure how the server performs when only five hundred Vusers deposit cash, you can deactivate (disable) the "withdraw" rendezvous, and instruct only five hundred Vusers to participate in the "deposit" rendezvous.

The following procedure outlines how to control load peaks on the server:

1. **Create the Vuser scripts, inserting the necessary rendezvous points.**
2. **Create a session step.**

   When you add a Vuser group to a session step, LoadRunner Tuning Module scans the group's associated script for the names of the rendezvous points and adds them to the list in the Rendezvous Information dialog box (**Session** > **Rendezvous**). If you create another Vuser group that runs the same script, the Console adds the new Vusers to the rendezvous and updates the list.

3 Set the level of emulated user load.

You determine the exact level of load by selecting the rendezvous points that will take part in the session step, and how many Vusers will participate in each rendezvous.

4 Set the attributes for the rendezvous (optional).

For each rendezvous you can set **Policy** attributes. For more information, see "Setting the Rendezvous Policy," on page 171.

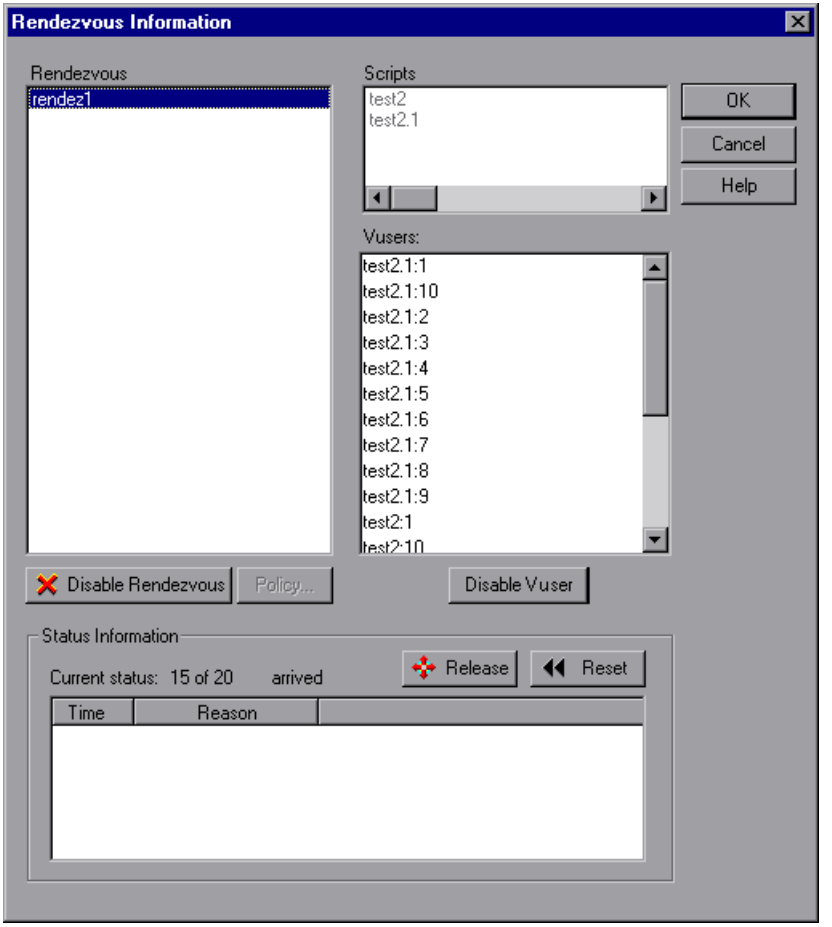
5 Run the session step.

Setting the Rendezvous Attributes

You can set the following rendezvous attributes from the Rendezvous Information dialog box (**Session** > **Rendezvous**):

- Rendezvous Policy
- Enabling and Disabling of Rendezvous Points
- Enabling and Disabling of Vusers

In addition, the dialog box displays general information about the rendezvous point: which script is associated with the rendezvous, and release history.



For information on manipulating the Vusers during step execution using the Release command, see Chapter 17, "Running an Execution Step."

Setting the Rendezvous Policy

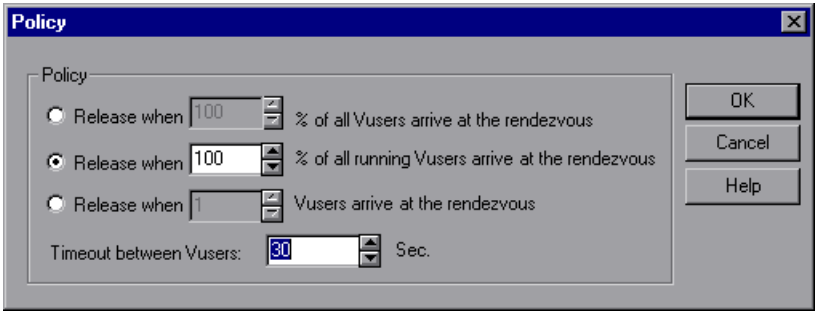
Setting the rendezvous policy determines how the Vusers handle a rendezvous point. You set the following policy attributes for each rendezvous:

release policy	sets how many Vusers will be released from a rendezvous at a time.
timeout	how long the Console waits before releasing Vusers from a rendezvous.

To set the rendezvous policy attributes:

1 Choose **Session** > **Rendezvous**. The Rendezvous Information dialog box opens.

2 Select a rendezvous from the Rendezvous box, and click the **Policy** button. The Policy dialog box opens.



3 In the Policy section, select one of the following three options:

➤ **Release when X% of all Vusers arrive at the rendezvous**: Releases the Vusers only when the specified percentage of all Vusers arrives at the rendezvous point.

Note: This option interferes with the scheduling of your session step. If you select this option, therefore, your session step will not run as scheduled.

➤ **Release when X% of all running Vusers arrive at the rendezvous**: Releases the Vusers only when the specified percentage of all Vusers running in the session step arrives at the rendezvous point.

➤ **Release when X Vusers arrive at the rendezvous**: Releases the Vusers only when the specified number arrives at the rendezvous point.

4 Enter a timeout value in the Timeout between Vusers box. After each Vuser arrives at the rendezvous point, waits up to the maximum **timeout** period you set for the next Vuser to arrive. If the next Vuser does not arrive within the **timeout** period, the Console releases all the Vusers from the rendezvous.

Each time a new Vuser arrives, the timer is reset to zero. The default **timeout** is thirty seconds.

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

Disabling and Enabling Rendezvous Points

You can temporarily disable a rendezvous and exclude it from the session step. By disabling and enabling a rendezvous, you influence the level of server load.

You use the Disable Rendezvous/Enable Rendezvous button in the Rendezvous Information dialog box, to change the status of a rendezvous.

To disable a rendezvous:

1. In the Rendezvous box, select the rendezvous you want to disable.
2. Click the **Disable Rendezvous** button. The button changes to **Enable Rendezvous** and the rendezvous becomes disabled.

To enable a rendezvous:

1. In the Rendezvous box, select the disabled rendezvous that you want to enable.
2. Click the **Enable Rendezvous** button. The button changes to **Disable Rendezvous** and the rendezvous becomes enabled.

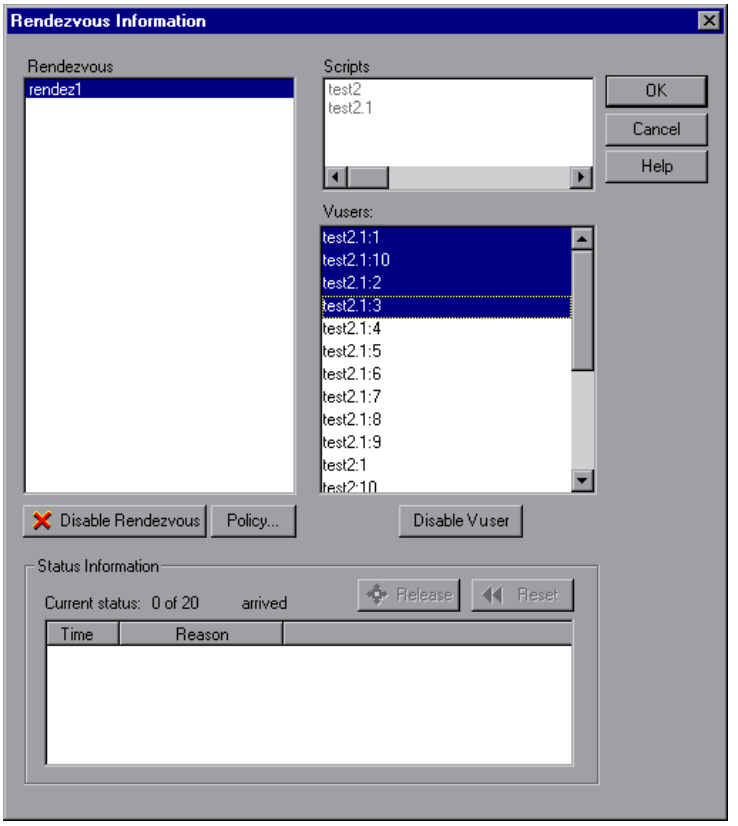
Disabling and Enabling Vusers at Rendezvous Points

In addition to disabling a rendezvous point for all Vusers in a session step, LoadRunner Tuning Module lets you disable it for specific Vusers. By disabling Vusers at a rendezvous, you temporarily exclude them from participating in the rendezvous. Enabling disabled Vusers returns them to the rendezvous. You use the Disable and Enable commands to specify which Vusers will take part in a rendezvous.

To disable a Vuser in a rendezvous:

1. In the Rendezvous box, select the rendezvous for which you want to disable Vusers.

2 In the Vusers box, select the Vuser(s) you want to exclude from the rendezvous. Select multiple Vusers using the CTRL key.



3 Click the **Disable Vuser** button below the Vusers box. The disabled Vusers change from black to gray and will not take part in the rendezvous.

To enable a Vuser, select it and click **Enable Vuser**.

Viewing Rendezvous Information

During and after a session step, you can view the rendezvous status in the Rendezvous Information dialog box. The following information is provided:

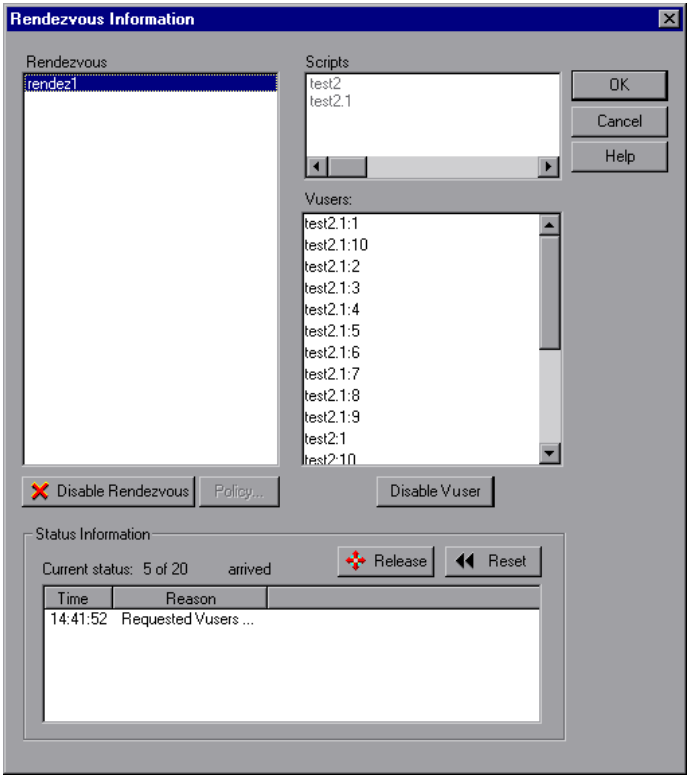
Time: The time at which the Vusers at the rendezvous point were released.

Reason: The reason the Vusers at the rendezvous point were released. The possible reasons are **Timeout** or **Arrived**.

Current Status: The number of Vusers that arrived at the rendezvous point, out of the total number of Vusers assigned to the rendezvous.

To view rendezvous information:

Select the rendezvous whose information you want to view. The rendezvous status is displayed in the Status Information section.



Understanding the Rendezvous Information Dialog Box

The Rendezvous Information dialog box enables you to view and modify the attributes of each rendezvous point in the session step.

Rendezvous: Displays the name(s) of the rendezvous point(s) in the session step.

➤ **Enable Rendezvous/Disable Rendezvous:** Enables or disables the selected rendezvous point(s) from participating in the session step.

Scripts: Lists the Vuser scripts that are associated with the rendezvous point(s).

Vusers: Lists the Vusers that are associated with the rendezvous point(s).

➤ **Enable Vuser/Disable Vuser:** Enables or disables a Vuser from taking part in the rendezvous.

Policy: Opens the Policy dialog box, enabling you to set how many Vusers are released from a rendezvous at a time, as well as the amount of time the Console waits before releasing Vusers from a rendezvous.

➤ **Timeout:** Enter the timeout value (in seconds). After each Vuser arrives at the rendezvous point, LoadRunner Tuning Module waits up to the number of timeout seconds specified for the next Vuser to arrive. If the next Vuser does not arrive within the timeout period, the Console releases all the Vusers from the rendezvous. Each time a new Vuser arrives, the timer is reset to zero. The default timeout is thirty seconds. You set a timeout for each rendezvous point.

Status Information

➤ **Current Status:** Displays the number of Vusers that arrived at the rendezvous point out of the total number of Vusers assigned to the rendezvous.

➤ **Time:** Displays the time at which the rendezvous was released.

➤ **Reason:** Displays the reason for the release of the Vusers from the rendezvous point. The possible reasons are **Timeout** or **Arrived**.

- **Release:** Releases all Vusers currently waiting at the selected rendezvous point. If you want the session step to continue even though all the Vusers did not reach the rendezvous, click this button.
- **Reset:** Resets the Status Information, removing the information currently displayed.

16

Preparing to Run a Session Step

Before you run a session step, you specify a location for the session step results and other run-time related settings.

This chapter describes:

- Specifying a Results Location
- Results Directory File Structure
- Collating Results

About Preparing to Run a Session Step

Before you run a session step, you need to specify the location of the results (mandatory), assign a name to the results, schedule the session step, and provide session step summary information. In addition, you can specify the applications to invoke at the start of a session step.

Although most of the pre-session step settings are optional, by using them you can enhance the testing process. These values are session step specific—you can set different values for each LoadRunner Tuning Module session step.

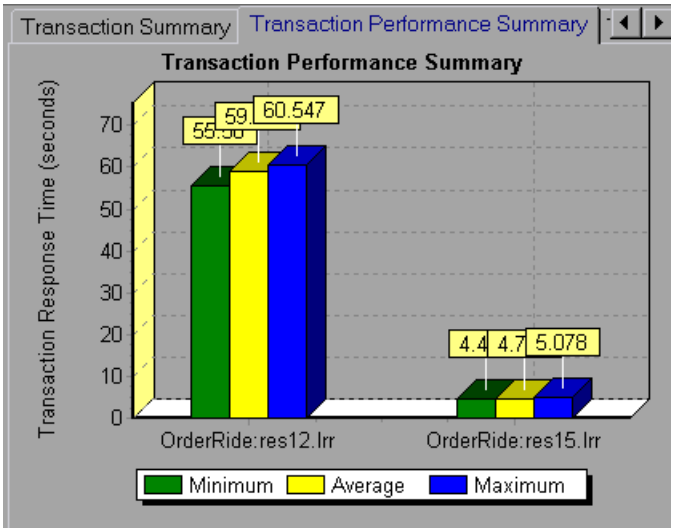
For information on one-time configuration settings such as timeout, output, and quotas, see Chapter 14, "Configuring Session Steps."

Specifying a Results Location

When you run a session step, by default the run-time files are stored locally on each load generator. After the session step, the results are collated together and processed on the Console machine. Alternatively, you can instruct LoadRunner Tuning Module to save the results on a shared network drive. For information about specifying a file storage method, see the Run-Time File Storage settings in Chapter 14, "Configuring Session Steps."

LoadRunner Tuning Module allows you to give descriptive names to each result set. This is especially useful for cross results analysis, in which LoadRunner Tuning Module superimposes the results of several session step runs in a single graph and lets you compare the results of multiple session step runs. The descriptive graph names enable you to distinguish between the results of the multiple runs.

In the example below, the results of two session step runs are superimposed. The result sets are *res12*, and *res15*.

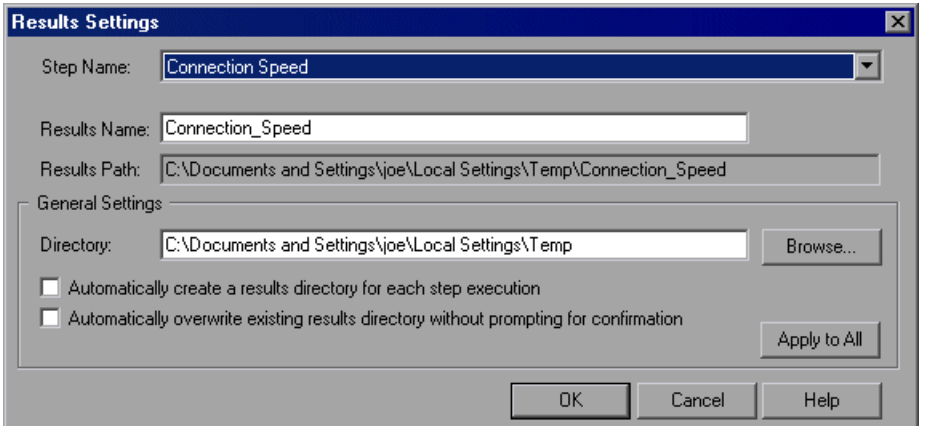


For more details on cross result graphs, refer to the *Analysis User's Guide*.

You can specify separate result settings for each session step.

To specify where results are stored:

1 Choose **Results** > **Results Settings**. The Results Settings dialog box opens.



2 Enter values in the Step Name, Results Name, and Directory fields, or accept the default values where relevant.

3 Select the appropriate check box for subsequent executions: **Automatically create a results directory for each session step execution** or **Automatically overwrite existing results directory without prompting for confirmation**.

4 To apply the settings specified in the dialog box to all the steps in your session, click **Apply to All**.

5 Click **OK** to save the results directory setting.

Understanding the Results Settings Dialog Box

The Results Settings Dialog Box lets you set the location in which the Console saves session step results.

Step Name: Shows the step to which the settings apply. To specify result settings for a different step, choose the step from the list in the box.

Results Name: Specify a name for the results, or accept the default name. By default, LoadRunner Tuning Module uses the step name as a basis for the results name. For example, if the step name is "Connection Speed", LoadRunner Tuning Module adds an underscore and uses

"Connection_Speed" as the results name. Avoid using the same name with different paths, since the names will appear identical on the graphs.

Results Path: Displays the full path to the subdirectory containing the step results. Using the default results name or the one you specified, the Console creates a subdirectory within the results directory. All results from this step are saved within this subdirectory.

Directory: Shows the full path to the results directory. Click **Browse** to specify a different path, or enter the new path in the box. If you are using the default file storage setting (local machine), specify a directory in which to store all of the collated results after the session step run. If you specified a shared network drive as the file storage method, specify the directory to which Vuser groups should write during session step execution. The Console creates a subdirectory within the results directory. All results are saved within this subdirectory.

Automatically create a results directory for each step execution: Instructs LoadRunner Tuning Module to create a unique results directory for each session step execution. By default, the result names are *res1*, *res2*, *res3*, etc.

Automatically overwrite existing results directory without prompting for confirmation: Instructs LoadRunner Tuning Module to automatically overwrite previous result sets, without prompting the user.

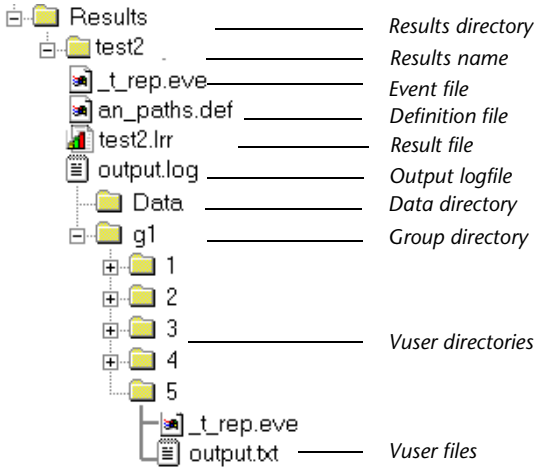
Apply to All: Applies the settings specified in the dialog box to all the steps in your session.

Results Directory File Structure

When you set the results directory, you also specify a results name. LoadRunner Tuning Module creates a subdirectory using the results name, and places all of the data it gathers in that directory. Every set of results contains general information about the session step in a result file (*.lrr*) and an event (*.eve*) file.

During session step execution, LoadRunner Tuning Module also gathers data from each Vuser and stores it in an event file *_t_rep.eve* and an output file *output.txt*. LoadRunner Tuning Module creates a directory for each group in

the session step and a subdirectory for each Vuser. A typical result directory has the following structure:



- *t_rep.eve* in the main result directory contains Vuser and rendezvous information.
- **.def* are definition files for graphs that describe the online and other custom monitors.
- *results_name.lrr* is the LoadRunner Tuning Module Analysis document file.
- *output.log* contains output information about the session step generated during test execution.
- The *Data* directory contains the database created by the Analysis (from the results files).
- *g1* is a group directory. A separate directory exists for each Vuser group that runs in the session step. Each group directory consists of Vusers subdirectories.
- *t_rep.eve* in each Vuser directory contains transaction information.
- *output.txt* in each Vuser directory contains output information generated during replay.

When you generate analysis graphs and reports, the LoadRunner Tuning Module Analysis engine copies all of the session step result files (*.eve* and *.lrr*) to a database. Once the database is created (and stored in the *Data* directory), the Analysis works directly with the database and does not use the result files.

For information on LoadRunner Tuning Module Analysis, refer to the *Analysis User's Guide*

Collating Results

When you run a session step, by default all Vuser information is stored locally on each load generator. After session step execution, the results are automatically collated or consolidated—results from all of the load generators are transferred to the results directory. You set the location of the results directory as described in "Specifying a Results Location," on page 180.

Note: If you have selected to store all the session step results directly to a shared network drive, then collation of the results is not required. See "About Configuring a Session," on page 157 for details on changing how results are stored.

To disable automatic collation and clear the check mark adjacent to the option, choose **Results** > **Auto Collate Results**. To manually collate results, choose **Results** > **Collate Results** > **Collate**. The Collating Files dialog box opens, displaying the progress of result and log file collation from each load generator. To stop collating the results and close the dialog box, click **Stop** and then **Close.** To resume collating the results, select **Results** > **Collate Results** > **Continue stopped collation**.

Note: You can choose to disable log file collation. For more information, see "Options - General Settings," on page 760.

The log and result directories are only deleted from a load generator once LoadRunner Tuning Module successfully collates the results from the machine. You can therefore close the Console after saving a session step, and collate the results once you reopen the session step in the Console.

If collation fails due to a lack of disk space, select **Results > Collate Results > Recollate**. LoadRunner Tuning Module attempts to collate the results again.

Before generating the analysis data, LoadRunner Tuning Module automatically collates the results if they have not previously been collated.

Note: If you enabled the **Auto Load Analysis** option in the Results menu, the Analysis may open during a lengthy collation process, displaying Analysis summary data.

Part III

Executing a Tuning Session

17

Running an Execution Step

When you tune a session, LoadRunner Tuning Module simulates your environment and measures the system's performance.

This chapter describes:

- Running an Execution Step
- Continuing With Subsequent Steps
- Adding Vusers to a Running Execution Step
- Viewing and Controlling Vusers

About Running an Execution Step

When you run an Execution session step, the Vusers are assigned to their load generators and execute their scripts. During session step execution, LoadRunner Tuning Module:

- records the durations of the transactions you defined in the scripts
- performs the rendezvous included in the scripts
- collects error, warning, and notification messages generated by the Vusers

You can run an entire Execution step unattended, or you can interactively select the Vusers that you want to run. When the session step starts running, the Console first checks the session configuration information. Next, it invokes the scripts that you selected to run with the step. Then, it distributes each script to its designated load generator. When the Vusers are ready, they start executing their scripts.

While the step runs, you can monitor each Vuser; view error, warning, and notification messages generated by the Vusers; and stop both Vuser groups and individual Vusers. You can instruct LoadRunner Tuning Module to allow an individual Vuser or the Vusers in a group to complete the iterations they are running before stopping, to complete the actions they are running before stopping, or to stop running immediately. For more information, see "Configuring Session Run-Time Settings" on page 158.

You can also activate additional Vusers while the step is running, using the Run/Stop Vusers dialog box. For more information, see "Adding Vusers to a Running Execution Step" on page 196.

The session step ends when all the Vusers have completed their scripts, when the duration runs out, or when you terminate it.

If you specified subsequent steps for execution (see "Specifying Step Execution Order" on page 66), depending on whether or not the current step was successful, LoadRunner Tuning Module tells you the name of the next step and gives you the option of executing it immediately.

Note: When creating and running a script in VuGen, the full browser is used. This differs from a test run in the Console, where only the browser basics are used. There may be occasions when a test passes its run in VuGen, but fails when it is run in the Console. Before running a session in the Console with multiple Vusers, run a single Vuser to ensure the script is bug-free.

Running an Execution Step

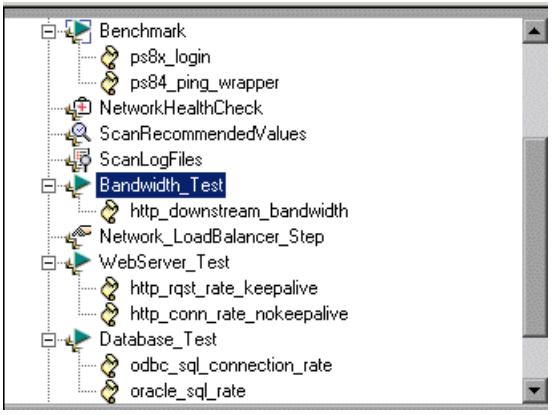
You can run all the Vusers specified by the session step settings, or you can select the number of Vusers that you want to run. Note that when you run your Execution step, LoadRunner Tuning Module runs the Vusers as soon as they reach the READY state.

The following section describes how to run an entire Execution step. For information on how to manipulate individual Vusers, see "Viewing and Controlling Vusers" on page 198.

To run an Execution step:

1 Open an existing session or create a new one.

2 In the Session tree, select the step that you want to execute.



The default monitors are displayed.

If the monitors displayed with the step were not changed since the step was created from a cartridge, the Console displays the cartridge's default monitors. If the step was used with other monitors and saved, the Console displays the monitors that were displayed when the step was saved.

3 Click the**Execute Execute** button in the Session tab's toolbar. The Console executes the step.

You can perform the following actions:

Add Vusers to a Running Step: Opens the Run/Stop Vusers dialog box, enabling you to activate or deactivate additional Vusers. For more information, see "Understanding the Run/Stop Vusers Dialog Box" on page 196.

When a step includes more than one script, you can either specify what percentage of the total number of Vusers should execute each script, or specify the number of Vusers. You can start or stop a specified number of Vusers, or a percentage of the total.

Note: The Console begins running the step at the time specified in the Session Step Start dialog box (**Session > Start Time**).

Next Step: Loads the next session step.

Gradual Stop: Instructs the Console to stop running the step. If you selected the **Stop immediately** option in the Run-Time Settings tab of the Options dialog box, all of the Vusers in the step move to the EXITING status. If you selected the **Wait for the current iteration to end before exiting** or **Wait for the current action to end before exiting** options in the Run-Time Settings tab of the Options dialog box, the Vusers in the step move to the GRADUAL EXITING status and exit the step gradually.

Stop the Step: Instructs the Console to stop running the step immediately.

Status bar: Displays a synopsis of the running step including step status, number of active Vusers, elapsed time, passed transactions, failed transactions, errors, and alerts. When you click the number of failed or passed transactions, the Transactions window displays statistics on the transactions that have taken place in the step. When you click the number of errors or alerts, the appropriate Output window is displayed. When running a goal-oriented step, the measurement used for the goal appears following the elapsed time.

System Topology pane: Displays the components defined in the business process that you are testing. Right-click on the topology diagram to display the zoom options. You can open the System Topology window by double-clicking the white space in the topology pane. For information about the System Topology window, see “Understanding the System Topology Window” on page 30.

Add/Remove Graphs: Displays the Select Online Graphs dialog box. See “Opening Online Monitor Graphs” on page 295.

When you first open or create an execution step, two graphs are displayed by default. To display only one graph, double-click the desired graph. To return to the previous view, double-click the graph again. Below the graphs, a legend displays the statistics of the selected graph: color, scale, measurement/status, machine, maximum, minimum, average, standard deviation, and last.

You can perform the following graph actions by using the right-click menu of the graph:

- Close graph
- Duplicate graph
- Specify the number of graphs to display
- Open a new graph
- Overlay graphs
- Export graph to HTML
- Freeze/Release graph
- Configure graphs
- Open the Alerts Definition page for the measurement

You can perform the following actions on a measurement on the graph by right-clicking it:

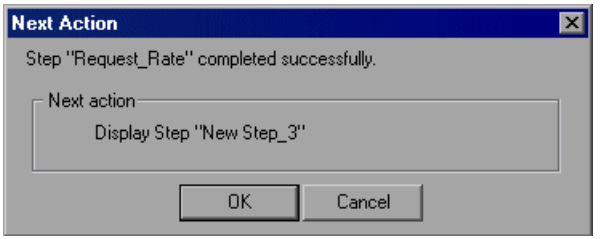
- Show only the selected measurement
- Show/Hide the measurement
- Configure the measurement
- View the measurement's description
- Open the Alerts Definition page for the measurement

View and Control Vusers: Opens the Vuser Management window, which displays information about the groups of Vusers in the session step. For more information, see "Understanding the Vuser Management Window" on page 199.

Continuing With Subsequent Steps

When you defined your step, you had the option of specifying the steps that should be executed after the current step completes execution (see "Specifying Step Execution Order" on page 66).

If you specified the subsequent step or steps, LoadRunner Tuning Module displays the Next Action dialog box when the current step completes execution.



Understanding the Next Action Dialog Box

The Next Action dialog box tells you whether the step succeeded or failed, and displays the name of the appropriate subsequent step.

If the step ended as the result of an alert that stopped all the Vusers, the step is considered to have failed.

If the step ended in any other way, the step is considered to have succeeded.

Display Step <*step name*>—Name of the specified subsequent step.

Click **OK** to change the view to the Session tab, with the focus on the specified subsequent step. This does not execute the step.

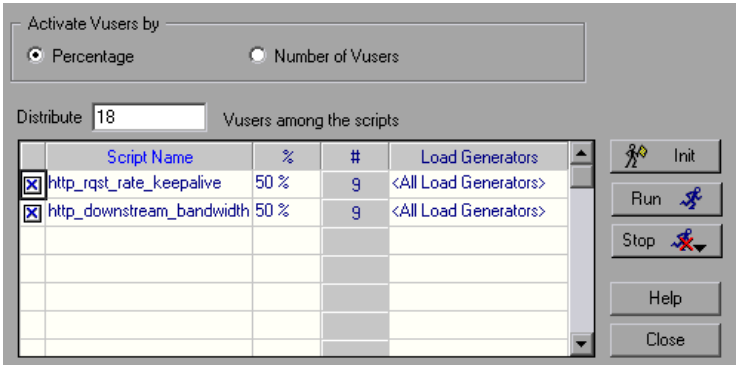
Click **Cancel** to leave the focus on the step that completed execution.

Adding Vusers to a Running Execution Step

You can add Vusers to a running Execution step to manually ramp up the load and see how your system performs when users are added. You can either distribute Vusers among scripts by percentage, or specify the number of Vusers for each script.

To add Vusers to a running Execution step:

1 Click the **Add Vusers to a Running Step** button in the top left section of the Session tab. The Run/Stop Vusers dialog box opens.



2 Specify the Vusers that you want to add to the step.

Understanding the Run/Stop Vusers Dialog Box

The Run/Stop Vusers dialog box lets you activate or deactivate additional Vusers.

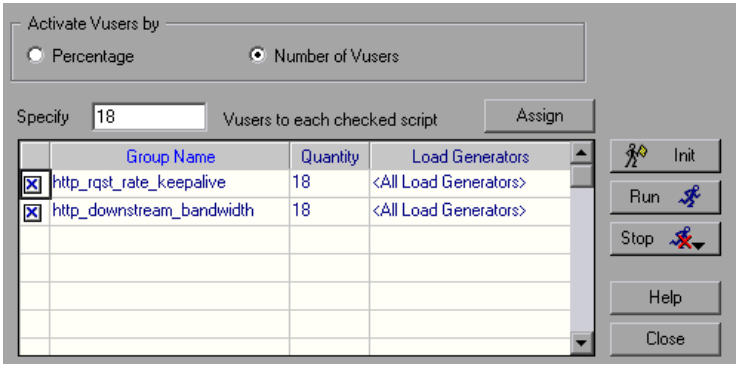
When a step includes more than one script, you can either specify what percentage of the total number of Vusers should execute each script, or specify the number of Vusers. You can start or stop a specified number of Vusers or a percentage of the total.

Activate Vusers by... Choose the appropriate radio button to specify whether you want to specify the number of Vusers for each script or specify the percentage of the total.

Percentage: Clicking this radio button displays the **Distribute ... Vusers among the scripts** box.

Distribute ... Vusers among the scripts: Enter the number of Vusers you want to distribute by percentage among the checked Vuser scripts. LoadRunner Tuning Module automatically distributes the number of Vusers you entered. The % column indicates the percentage of Vusers distributed to each Vuser script. The # column indicates the number of Vusers distributed to each Vuser script. The Load Generators column indicates the load generators on which the Vusers will be run. Note that if more than one load generator is defined for a script, the added Vusers are proportionally distributed among the defined load generators. To manually change the percentage of Vusers assigned to a script, enter the new percentage in the script's % column. LoadRunner Tuning Module automatically adjusts the percentages of the other scripts.

Number of Vusers: Clicking this radio button displays the **Specify ... Vusers to each checked script** box.



This displays the Vusers by their quantity distribution. To manually change the number of Vusers assigned to a script, enter the new number in the script's Quantity column.

Specify ... Vusers to each checked script: Allows you to specify the number of Vusers to run each script.

To disable a Vuser script, clear the check box to the left of the script name. Note that a script will automatically appear disabled if it is disabled in the Design view.

Note: If you disable a script, no Vusers will be distributed to it. However, 100 percent of the Vusers will not be distributed among the remaining scripts, unless you define a 0 percent value for the disabled script.

Init: Distributes the Vusers that you added to their designated load generators so that they are ready to execute their scripts. The Console first initializes the Vusers in your step that were not yet run, on the load generator(s) defined in the Run/Stop Vusers dialog box. It then adds additional Vusers, as required, to reach the quantity defined in the current dialog box.

Run: Runs the number of Vusers you specified. The Console first runs the Vusers in your step that were not yet run, on the load generator(s) defined in the current dialog box. It then adds additional Vusers, as required, to reach the quantity defined in the current dialog box.

Stop: Stops the Vusers that are running on the load generator(s) defined in the current dialog box. The Console stops the Vusers according to the settings you defined in the Run-Time Settings tab of the Options dialog box.

Viewing and Controlling Vusers

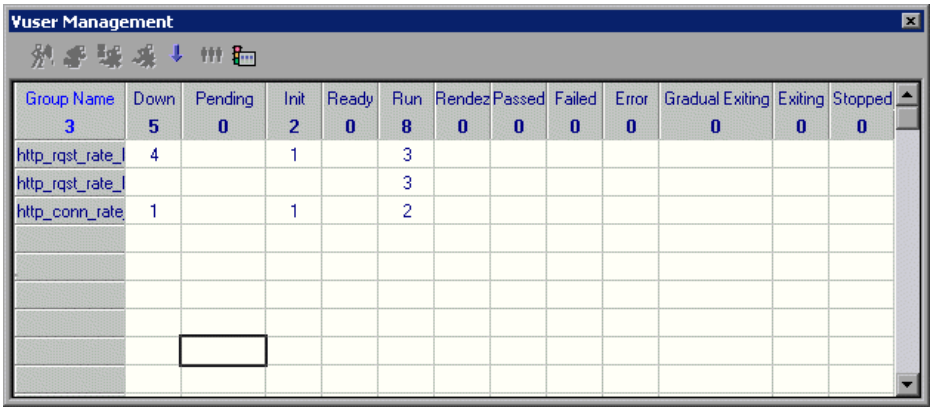
LoadRunner Tuning Module allows you to see different views of the Vusers running in a step.

You can also manipulate individual Vusers. This includes initializing, running, and stopping individual Vusers.

To view Vuser groups:

➤ Click **View and control Vusers** to open the Vuser Management—Groups window. (If LoadRunner Tuning Module opens the Vuser Management—

Groups by Load Generator window instead, click the **Show Groups**button in the task bar to display the Vuser Management—Groups window.)



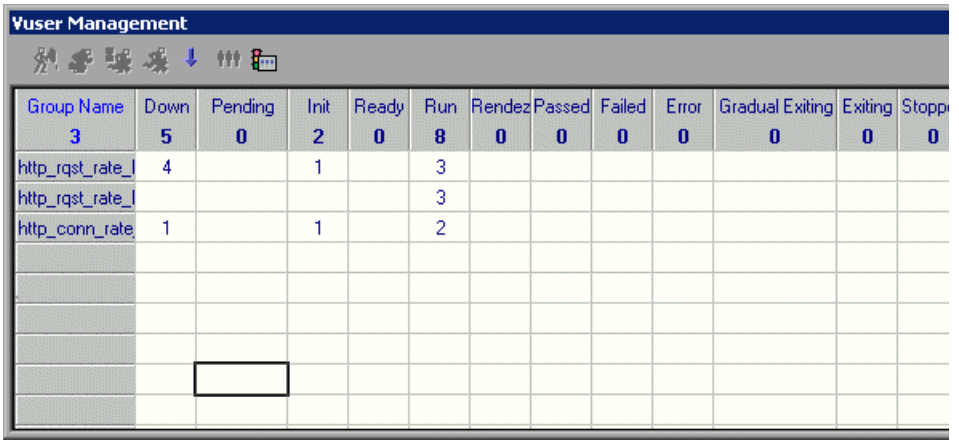
Each row of the table lists a group of Vusers in the session step. The columns show how many of the group's Vusers have a particular status. You can do the following:

- Show the Vusers in each group. You can filter or sort the Vusers.
- Start or stop a Vuser.
- View a Vuser executing its assigned script.
- View the script log.

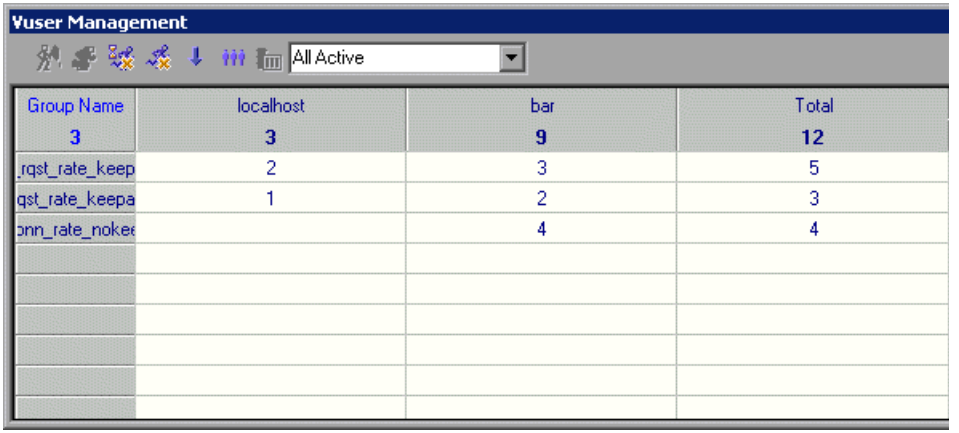
Understanding the Vuser Management Window

The Vuser Management window displays information about the groups of Vusers in the session step and allows you to control Vusers, including: initializing, starting, stopping, resetting, enabling and disabling Vusers. Note that each column heading contains the total number of Vusers or groups displayed in the column.

Show Vuser Status By Group: Click this button to view the Vusers in each group, broken down by status (for example, Down, Pending, and so on):



Show Groups by Load Generator: Click this button to view the number of Vusers that each group is running on each load generator.

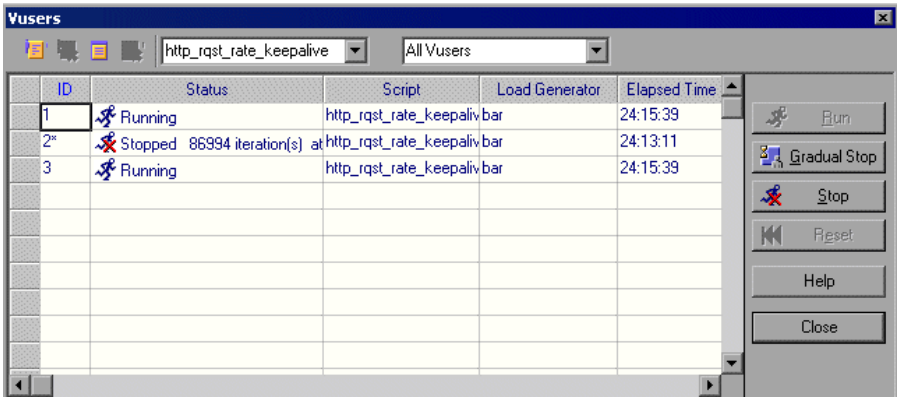


From the drop-down list, specify which Vusers you want to display (that is, Vusers with a particular status).

Each row of the table lists a group of Vusers in the session step. The columns show how many Vusers in each group, with the status you specified, are running on each load generator.

To control an individual Vuser:

➤ In the Groups window or the Groups by Load Generator window, double-click a group, or select a group and click the **Show Vusers in Group** button. The Vusers in Group dialog box opens, showing the ID, Status, Script, Load Generator, and Elapsed Time (since the beginning of the session) for each of the Vusers in the group.



Note: To choose a different Vuser group, select it from the left list box.

Understanding the Vusers in Group Dialog Box

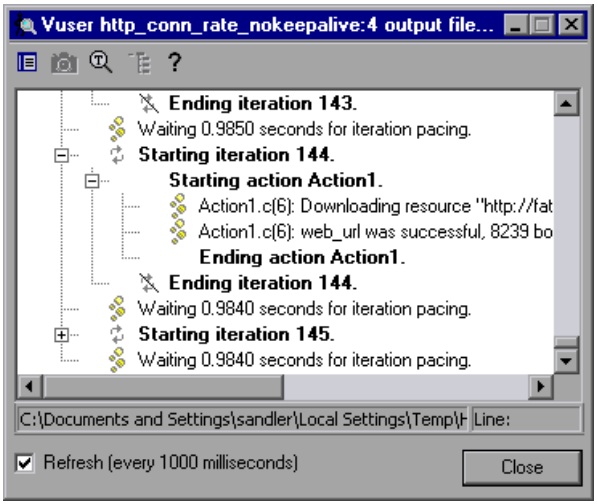
The Vusers in Group dialog box lists a group's Vusers, and shows you the following information about each: ID, Status, Script, Load Generator, and Elapsed Time (since the beginning of the session). It also allows you to control each Vuser's activity.

You can view different Vuser groups by selecting the desired group from the left list box. Use the right list box to specify whether you want to display all the Vusers in a group or just those that have a specific status.

Show the Selected Vusers: Opens a Run-Time Viewer for each selected Vuser, allowing you to see the Vuser executing the script. See "Understanding the Run-Time Viewer" on page 204.

Hide the Selected Vusers: Closes the open Run-Time Viewers .

Show Vuser Log: Displays a log containing run-time information about the Vuser. A script log opens.



The log information is refreshed, by default, every 1000 milliseconds.

For more information on the script log, see page 217.

Hide Vuser log: Closes the script log. Alternatively, you can click **Close** in the log window.

ID: Displays the Vuser's ID number.

Status: Displays the Vuser's status. The possible statuses are:

Status	Description
DOWN	The Vuser is down.
PENDING	The Vuser is ready to be initialized and is waiting for an available load generator, or is transferring files to the load generator. The Vuser will run when the conditions set in its scheduling attributes are met.
INITIALIZING	The Vuser is being initialized on the remote machine.
READY	The Vuser already performed the init section of the script and is ready to run.

Status	Description
RUNNING	The Vuser is running. The Vuser script is being executed on a load generator.
RENDEZVOUS	The Vuser has arrived at the rendezvous and is waiting to be released by LoadRunner Tuning Module.
DONE.PASSED	The Vuser has finished running. The script passed.
DONE.FAILED	The Vuser has finished running. The script failed.
ERROR	A problem occurred with the Vuser. Check the Status field on the Vuser dialog box or the output window for a complete explanation of the error.
EXITING	The Vuser has finished running or has been stopped, and is now exiting.
STOPPED	The Vuser stopped when the Stop command was invoked.

Script: Displays the script run by the Vuser.

Load Generator: Displays the load generator machine on which the Vuser is running.

Elapsed Time: Displays the amount of time that has elapsed in the step since the Vuser began running.

Run: Instructs the Console to begin running the selected Vuser(s). Alternatively, you can right-click the Vuser and click **Run Vuser/s.** If you run a Vuser group in the DOWN or ERROR state, LoadRunner Tuning Module initializes the group and then runs it.

Gradual Stop: Instructs the Console to complete the current iteration or action before stopping the Vuser. This option is only available when the Vuser is in the RUN state, if you selected the **Wait for the current iteration to end before exiting** or **Wait for the current action to end before exiting** options in the **Run-Time Settings** tab of the Options dialog box.

Stop: Instructs the Console to stop running the Vuser immediately.

Reset: Resets the status of the selected Vuser(s) to DOWN. Alternatively, you can right-click the Vuser and click **Reset**.

The following additional right-click options are available:

➤ **Run-Time Settings:** Displays the Run-Time Settings tab in the Options dialog box. For details, see "Understanding the Options - Run-Time Settings Tab" on page 159.

➤ **View Script:** Invokes VuGen to show you the script.

➤ **Initialize Vuser/s:** Distributes the Vusers in the session step to their designated load generators so that they are ready to execute their script. The Vuser group's status changes from DOWN to PENDING to INITIALIZING to READY. If a Vuser fails to initialize, its status changes to ERROR.

By initializing all of the Vusers in a group before running them, you can ensure that they all begin executing the step at the same time.

➤ **Pause:** Temporarily pauses the Vuser's execution. The status of the Vuser changes from RUNNING to PAUSED.

Note: Pausing a Vuser affects its transaction response time.

➤ **Filter Vusers:** Displays only Vusers of the type you specify. Right-click in one of the columns, select **Filter Vusers**, and select the filters you want to apply. Alternatively, you can select the filter option you want to use from the right-hand filter selector at the top of the dialog box.

➤ **Sort Vusers:** Sorts the listed Vusers by ID, status, script, load generator or elapsed time. Right-click in one of the columns, select **Sort Vusers**, and select the desired sort criteria.

Understanding the Run-Time Viewer

The Run-Time Viewer displays snapshots of the pages returned to a Vuser during replay. The Run-Time Viewer does not function as a browser, so the displayed images are snapshots and do not present all aspects of the replay.

Options: These menu items allow you to select the types of controls to display.

View: These menu items allow you to open various toolbars and views.

18

Viewing Vusers During Execution

During a session, you can view the actions that are performed by Vusers.

This chapter describes:

- Monitoring Vuser Status
- Viewing the Output Window
- Viewing the Script Log
- Viewing the Agent Summary

About Viewing Vusers During Execution

LoadRunner Tuning Module lets you view Vuser activity during a session:

- On the Console load generator machines, you can view the Output window, monitor Vuser performance online, and check the status of Vusers executing the session.
- On remote machines, you can view the Agent summary with information about the active Vusers.

Monitoring Vuser Status

During session execution, you can use the Vuser Management window in the Execute view to monitor the actions of all the Vusers and Vuser groups in the session.

When you click **Vuser Management**, the Vuser Management window shows a list of all the groups in the step, and the number of Vusers of each group

that are currently in each state. The following table describes the possible Vuser states during a session.

Status	Description
DOWN	The Vuser is down.
PENDING	The Vuser is ready to be initialized and is waiting for an available load generator, or is transferring files to the load generator. The Vuser will run when the conditions set in its scheduling attributes are met.
INITIALIZING	The Vuser is being initialized on the remote machine.
READY	The Vuser already performed the init section of the script and is ready to run.
RUNNING	The Vuser is running. The script is being executed on a load generator.
RENDEZVOUS	The Vuser has arrived at the rendezvous and is waiting to be released by LoadRunner Tuning Module.
DONE.PASSED	The Vuser has finished running. The script passed.
DONE.FAILED	The Vuser has finished running. The script failed.
ERROR	A problem occurred with the Vuser. Check the Status field on the Vuser dialog box or the output window for a complete explanation of the error.
GRADUAL EXITING	The Vuser is completing the iteration or action it is running (as defined in Tools > Options > Run-Time Settings) before exiting.
EXITING	The Vuser has finished running or has been stopped, and is now exiting.
STOPPED	The Vuser stopped when the Stop command was invoked.

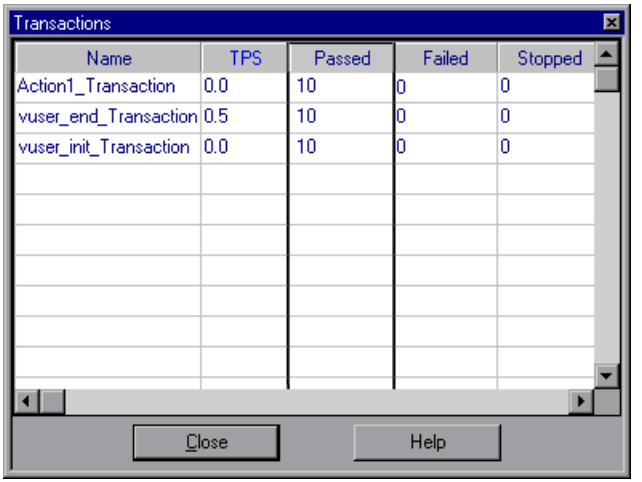
You can also view a synopsis of the running session in the Status bar below the toolbar.

Status Summary	Description
STEP STATUS	indicates whether the session is RUNNING or DOWN
ACTIVE VUSERS	indicates how many Vusers are being executed on a load generator machine
ELAPSED TIME	indicates how much time has elapsed since the beginning of the session
PASSED TRANSACTIONS	indicates how many transactions have been executed successfully
FAILED TRANSACTIONS	indicates how many transactions have been executed unsuccessfully
ERRORS	indicates how many problems have occurred with the Vusers
ALERTS	indicates how many alerts have been triggered
PROGRESS	indicates how close you are to the defined goals

Note: When running a goal oriented step, the measurement used for the goal appears following the elapsed time.

To view details of the transactions:

➤ The number of failed transactions in the toolbar is a link. Click the link to open the Transactions dialog box.



Understanding the Transactions Dialog Box

The Transactions dialog box lets you view details of individual transactions in a script.

Name: Transaction name.

TPS: Number of transactions per second.

Passed: Number of transactions that passed.

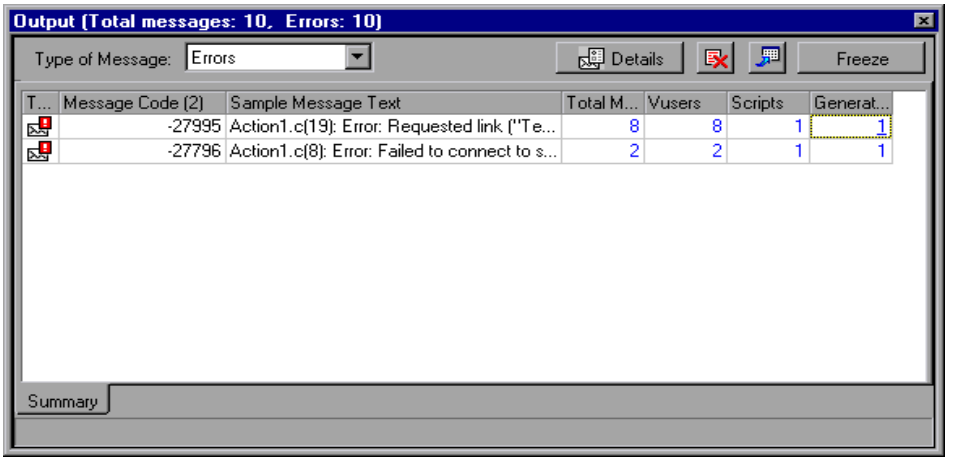
Failed: Number of transactions that failed.

Stopped: Lists the number of transactions that stopped before completion.

To view details of the errors:

➤ The number of errors in the toolbar is a link. Click the link or choose **View** > **Show Output** to display the Error log information in the Output

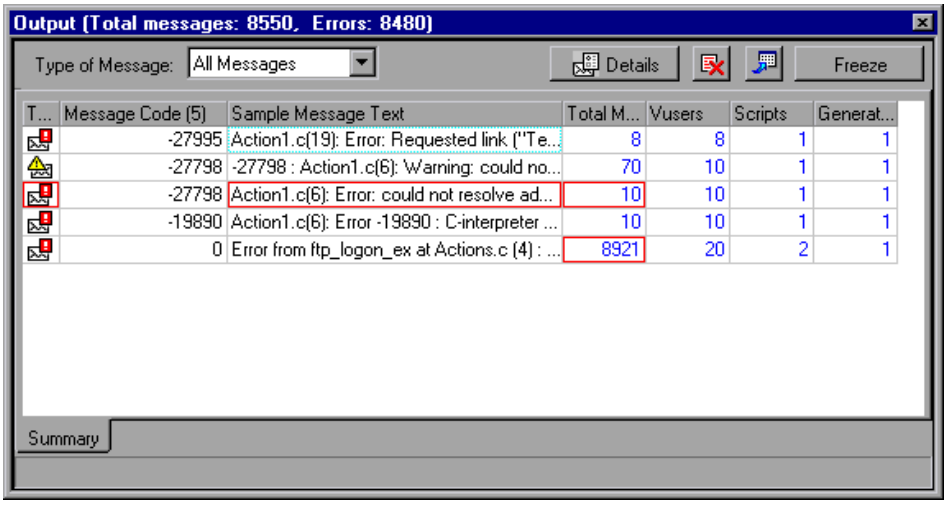
window. **Note:** If the Output window does not display errors, choose Errors from the Type of Message list.



For each type of error message code, the Output window lists a sample message text, the total number of messages generated, the Vusers and load generators that generated the code, and the scripts in which the errors occurred. To view details of the log information by message, Vuser, script, or load generator, click the link in the appropriate column. For more information on the Output window, see “Viewing the Output Window” on page 212.

Viewing the Output Window

While the session runs, the Vusers and load generators send messages to the Console. In addition, the Console generates alert messages. You can view all of these messages in the Output window.



Note: You can limit the number of messages in the Output window and set a deletion quota for the number of messages that will be overwritten. For more information, see Appendix C, "Working in Expert Mode."

Understanding the Output Window

The Output window displays information about the error, notification, warning, debug, and batch messages that Vusers and load generators send to the Console, and the alert messages generated by the Console itself, during step execution. Note that the total number of messages sent is displayed in the title bar.

LoadRunner Tuning Module clears the messages in the Output window at the start of each step execution, or when you reset a step.

To show the Output window, choose **View** > **Show Output**, or click either the Errors or the Alerts link in in the Status bar. To hide the Output window, choose **View** > **Show Output** or close the window.

Summary: This tab displays summary information about the messages sent during step execution. To view details of each message, Vuser, script, and load generator associated with an error code, click the blue link in the appropriate column. To sort the log information, click the appropriate column header. The messages are sorted in descending/ascending order.

Type of Message: Filters the output messages to display only certain message types. By default, all types of output messages are displayed, unless you specify the message type in this field or open the Output window by clicking on a link in the Status bar. Select one of the following filters:

- **All Messages:** Displays all message types.
- **Notifications:** Provide run-time information, such as messages sent using lr_output_message.
- **Errors:** Usually indicate that the script failed.
- **Warnings:** Indicate that the Vuser encountered a problem, but test execution continued
- **Debug Messages:** Will only be sent if you enable the debugging feature in Tools > Options > Debug Information (Expert Mode). See "Options - Debug Information Settings" on page 761 for more information.
- **Alert Messages:** Sent only if alerts were enabled during script execution. If you enabled alerts after script execution, run the step again to generate alerts.

Details: Displays the full text of the selected output message in the Output window's Detailed Message Text box.

Remove all messages: Clears all log information from the Output window.

Export the view: Saves the Output view to a specified file.

Freeze/Resume: Stops updating the Output window with messages. To instruct LoadRunner Tuning Module to resume updating the Output window, click the **Resume** button. Note that newly updated log information is displayed in a red frame.

Type: Displays an icon indicating the type of message received. The following icons are used:

Error Message

Warning Message

Notification Message

Debug Message

Alert Message

Message Code: Displays the code assigned to all similar messages. The number in parentheses indicates the number of different codes displayed in the Output window.

Sample Message Text: Displays an example of the text of a message with the specified code.

Total Messages: Displays the total number of sent messages with the specified code.

Vusers: Displays the number of Vusers that generated messages with the specified code.

Scripts: Displays the number of scripts whose execution generated messages with the specified code.

Generators: Displays the number of load generators that generated messages with the specified code.

Filtered: This tab appears after you drill down on a column, and displays a view by message, Vuser, script, or load generator. For example, if you drill down on the Vusers column, the Filtered tab displays all the messages with the code you selected, grouped by the Vusers that sent the messages.

Previous view/Next view: Enables you to move between the various drill-down levels.

[Type of Message Icon]: Displays an icon indicating the type of message filtered by the current Output view.

Active Filter: Displays the category or categories filtered by the current Output view.

Viewed by: Displays the name of the selected drill-down column.

Refresh: Adds new log information that arrived in the Output window to the Detailed tab view.

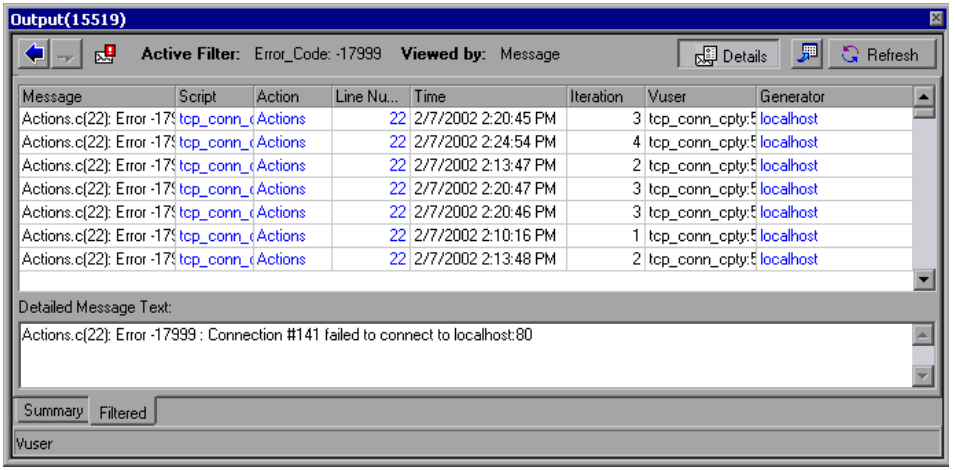
You can drill down further on the entries displayed in blue. Note that when you drill down on a Vuser, the Vuser log opens. When you drill down on a load generator, the Load Generators dialog box opens, displaying the load generator you selected. When you drill down on a script (or Action or Line Number), VuGen opens, displaying the script you selected..

Note: In the Output tab of the Options dialog box, you can configure the Show Vuser operation and set a deletion quota for the number of messages that can be generated before they are overwritten. The Output tab is available when Expert mode is enabled. For more information, see "Options - Output Settings" on page 763.

Viewing Log Information Details

You can view details of each message, Vuser, script, and load generator associated with an error code by clicking the blue link in the respective column. The Output window displays a drilled-down view by message, Vuser, script, or load generator in the Filtered tab.

For example, if you drill down on the Vusers column, the Output window displays all the messages with the code you selected, grouped by the Vusers that sent the messages.



Note that the message type, the message code, and the selected drill-down column, are displayed above the grid.

You can drill down further on the entries displayed in blue. Note that when you drill down on a Vuser, the Vuser log opens. When you drill down on a load generator, the Load Generators dialog box opens, displaying the load generator you selected. When you drill down on a script (or Action or Line Number), VuGen opens, displaying the script you selected.

Note: To limit the number of rows displayed when you drill down, open the *wlrun7.ini* file in any text editor, and located the following line:
MaxOutputUIRowsToShow=0
Change the 0 (no limit) to the number of rows you want to view.

When new messages arrive in the Output window, the Refresh button is enabled. Click **Refresh** to add the new log information to the Detailed tab view.

To move between the various drill-down levels, click the **Previous view** and **Next view** buttonsin the upper left-hand corner of the Output window.

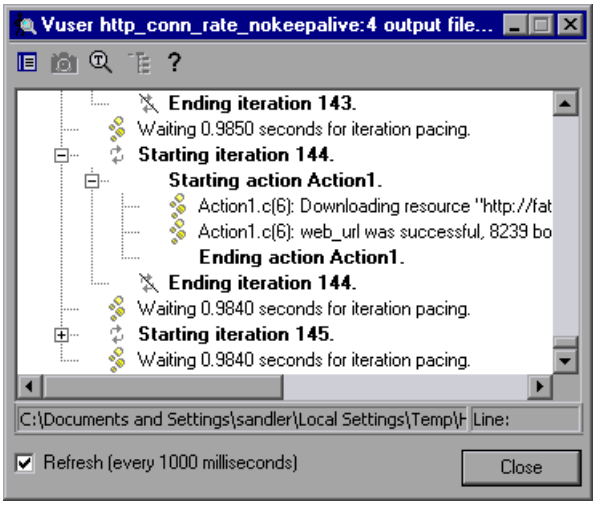
Viewing the Script Log

During session execution, you can view a log containing run-time information about each running Vuser.

To view the script log for a particular Vuser:

➤ In the Vusers dialog box, select the Vuser whose log you want to view, and click the **Show Vuser Log** button, or right-click the Vuser and select **Show**

Vuser Log.
The script log opens.



The log displays run-time information about the Vuser and is refreshed, by default, every 1000 milliseconds. To change the default refresh settings, see "Options - Output Settings" on page 763.

Note: If you disabled the logging feature in the Run-Time Settings' Log tab, the script log will be empty. If you selected the **Send messages only when an error occurs** option in the Log tab, the script log will contain output only if there are script errors.

➤ To disable the refreshing of this log, clear the **Refresh** check box.

➤ To view the information in text format, click the **Show Text View** button. To revert to the tree view, click the button again.

➤ If you are running a Web Vuser, and want to view a snapshot of the Web page where an error occurred, highlight the error in the Vuser log and click the **Display** button. Note that this option is only available for Vusers running on Windows load generators.

Note: To view a snapshot of the Web page where an error occurred, you must select the **Activate snapshot on error** option in the Internet Protocol > Preferences tab of the Run-Time Settings dialog box before running the session.

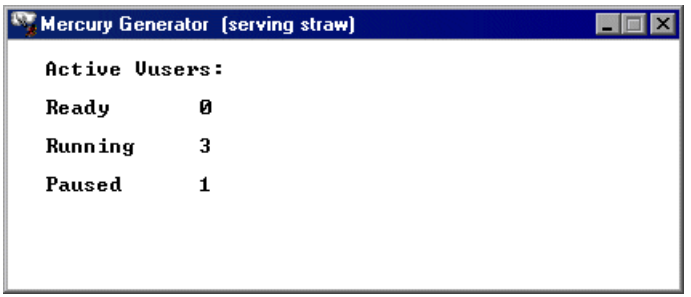
➤ To search the Vuser log for specific text, click the **Find Text** button, and enter the text you want to search for in the text box.

➤ To collapse the tree view, click the **Collapse Node** button. To revert to the expanded tree view, click the same button again.

➤ Click **Close** to close the script log.

Viewing the Agent Summary

When you run a session with non-GUI Vusers, the machine running the Vusers invokes an agent that controls the Vuser execution on that load generator. To view the Agent window during session execution, double-click the Mercury Generator Agenticon on the task bar. LoadRunner Tuning Module displays a summary of the Ready, Running, and Paused Vusers.



19

Viewing Step Run Progress

The Progress Dashboard shows you how close your benchmark step run came to accomplishing the improvements in performance that you defined for your session.

Note: If the benchmark step runs for less than 15 seconds, its progress is displayed as zero on the Progress Dashboard.

This chapter describes:

- Viewing the Progress Dashboard

About Viewing the Progress Dashboard

The Progress Dashboard displays the following information for the benchmark step:

- The session goals
- Information about the goal measurements in each run of the benchmark step
- How close each run came to achieving the goals

Viewing the Progress Dashboard

The Progress Dashboard allows you to view your progress after each run of the benchmark step.

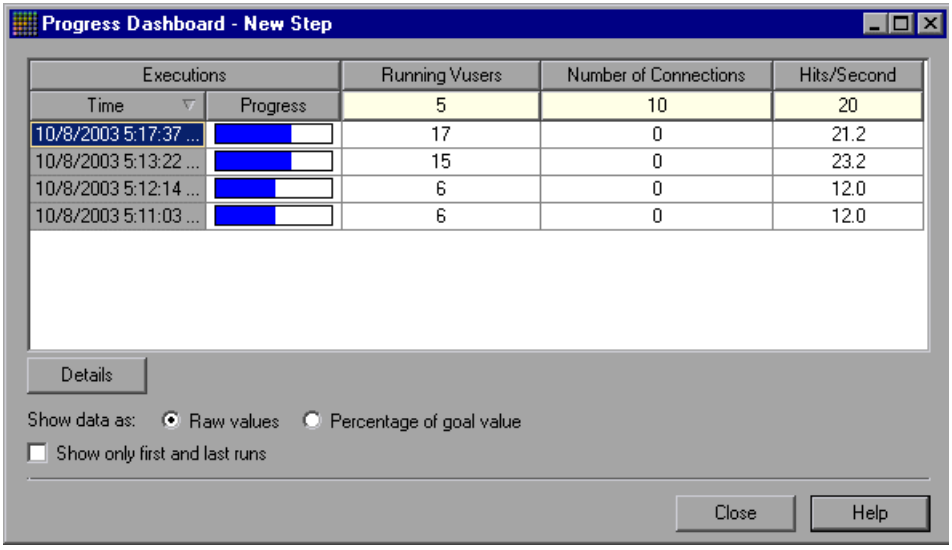
To view the Progress Dashboard, choose one of the following methods:

➤ Choose **Session** > **Progress Dashboard.**

➤ Click the Progress value in the Status bar.

➤ Click the root element of the Session tree.

The Progress Dashboard is displayed, showing you the defined goals and information about each step run.



Understanding the Progress Dashboard

The Progress Dashboard shows you how close each step run came to accomplishing the session goals. These are the goals defined in the Session Goals pane, visible when you click the root element of the Session tree (see "Viewing Session Information" on page 20). Each row in the table shows the measurements that were achieved in a step run.

To view general information about the step run, select it and click **Details**. The Step Details dialog box is displayed. (See "Understanding the Step Details Dialog Box" on page 243.)

Executions: Lists all the step runs.

- **Time:** When the step run began execution.
- **Progress:** How much of the goal was achieved during the step run. This value is calculated from all the goal values.

The top row of values shows the measurements that are defined for each goal in the Session Goal pane.

Show data as: The form in which you want the step run data to be displayed:

- **Raw values:** Displays the measurements as absolute values.
- **Percentage of goal value:** Displays the measurements as percentages of the defined goal measurements.

Show only first and last runs: Check to display only the first and last runs of the session.

20

Advanced Tuning Functions

In addition to Execution steps, the Console also provides a number of advanced tuning functions to measure network latency, scan log files for errors, and compare parameters with recommended values.

This chapter discusses:

- Network Health Check Step
- Scan Logs Step
- Scan Recommended Values Step

About Advanced Tuning Functions

The Tuning Module provides the following advanced functions, which you can use before or after running Execution steps:

- Network Health Check Step—measures the latency and number of hops between elements in your network
- Scan Logs Step—searches for error messages in the log files on your host machines
- Scan Recommended Values Step—compares the current values on a machine or service with the recommended values in LoadRunner Tuning Module's knowledge base, and highlights the values that differ from the recommended ones

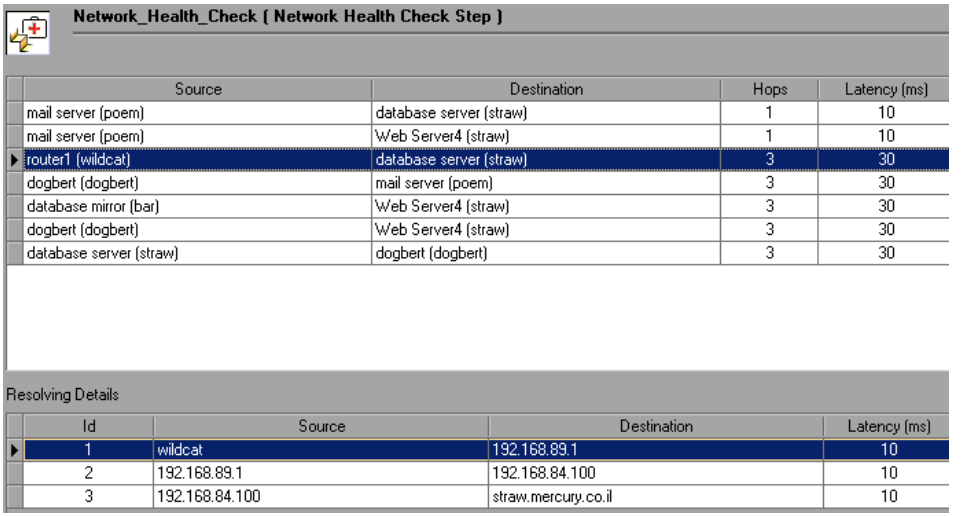
Network Health Check Step

A Network Health Check step gives you an indication of your network's overall efficiency.

You create a Network Health Check step by choosing **Session** > **Add Step** > **Network Health Check**, or by selecting an existing step in the session tree. The Network Health view consists of a table containing the pairs of components that are connected by lines in the Topology window, along with the latency and number of hops between the components. The table remains empty until you perform the step.

To perform a Network Health Check step:

1 Make sure that tuning agents are running on the host machines, and then click **Check Network Health**. LoadRunner Tuning Module performs the step and displays the results, as in the following example.



Note: The Network Health check view displays data only after you perform the network health check. When you initially create or display a Network Health Check step, the table of component pairs is empty.

Source and **Destination:** The logical and physical names of the two components.

Hops: The number of hops detected between the pair of components.

Latency (ms): The latency (in milliseconds) measured between the two components.

Resolving Details: Information about each hop between the pair of components selected in the table. This section displays the source and destination of each hop, and its latency. To view the resolving details, click anywhere in the relevant row.

Check Network Health: Measures latency and the number of hops between all the pairs of components connected in the Topology window, and displays the information in the table.

Scan Logs Step

Scan Logs steps locate error messages in various system logs on the host machine's services, helping you to pinpoint the services and applications that require tuning. These steps are useful for indicating errors on the host machine's services.

The log-scanning mechanism uses the tuning agents running on the host machines to scan the log files of applications that they can tune.

You can use the Custom Service feature to scan logs of applications that are not supported by tuning agents.

Defining a Scan Logs Step

When you create a Scan Logs step, LoadRunner Tuning Module displays the Log Scan view. This view allows you to scan the log files for error messages.

Note: The table displays values only if you ensure that tuning agents are running on the host machines, and perform a scan.

The upper section of the Scan Logs view consists of a table that displays information about log files scanned by LoadRunner Tuning Module.

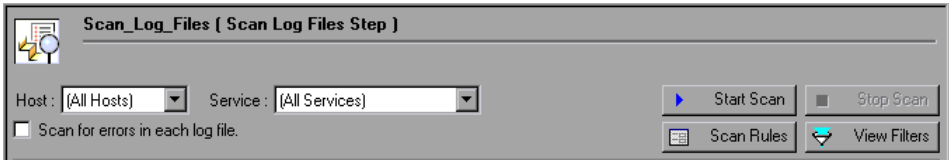
Scanning Log Files

Note: If you want to scan log files of an application that is not supported by the tuners, you first need to define the application as a custom service. See "Scanning Custom Service Log Files" on page 233.

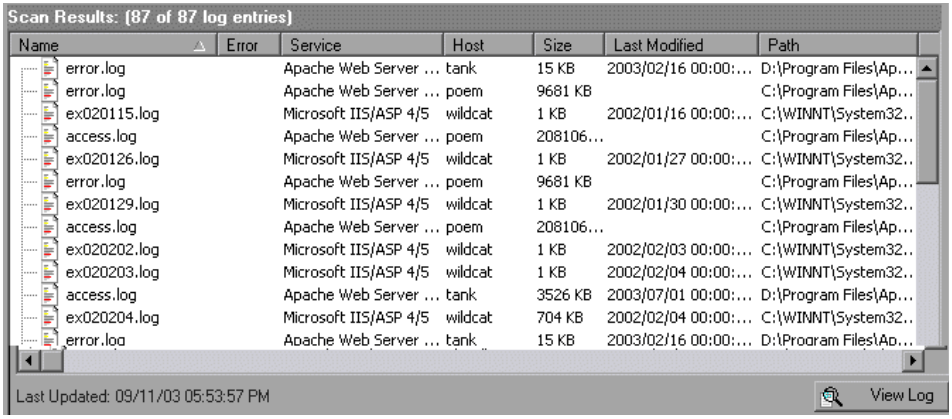
To view the hosts' log files:

1 Make sure that tuning agents are running on the host machines.

2 In the Host and Service lists, choose the hosts and services whose log files you want to scan.



3 Click **Start Scan**. The Console displays a list of the hosts' log files.



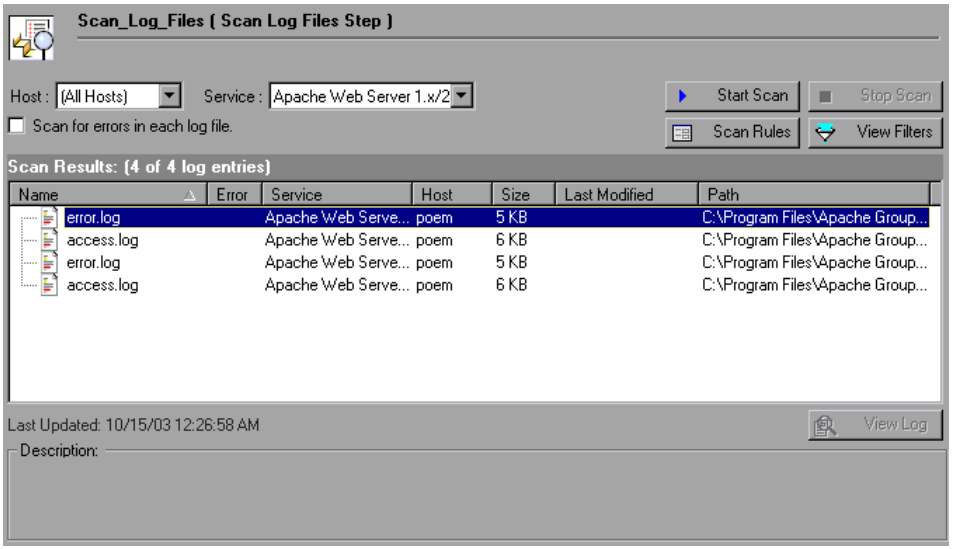
You can group the values as needed by dragging the column names.

To check the log files for error messages:

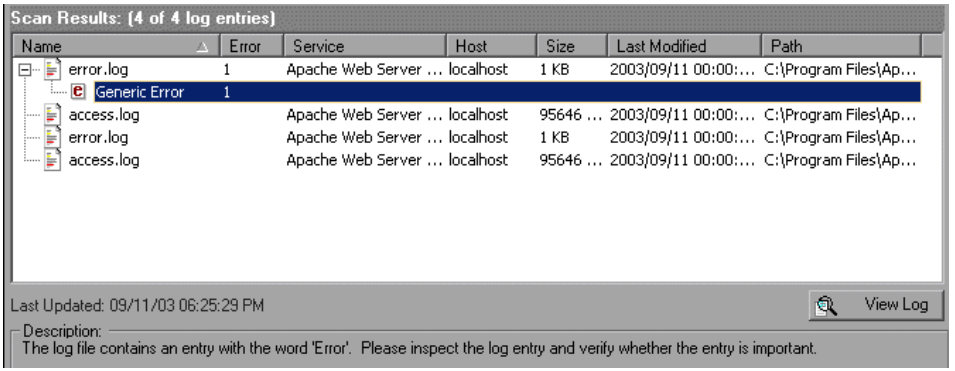
1 Check **Scan for errors in each log file**.

2 Choose the hosts and services.

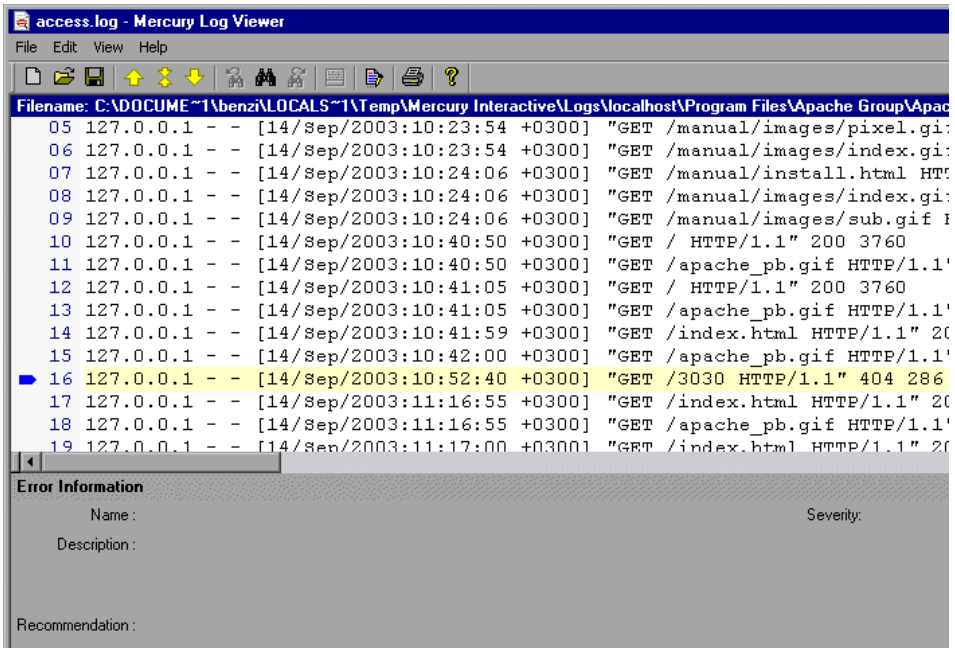
3 Click **Start Scan**. The Console displays a list of the hosts' log files. If an error message is detected in a log file, a plus sign appears to the left of the filename. In the following example, the first file in the list (error.log) is marked as containing an error message.



4 To view the error message, expand the filename and click the error message entry. A description of the error appears in the Description field.



5 To view the error message in the log file, click **View Log**. The log viewer opens. The error message entries are highlighted.



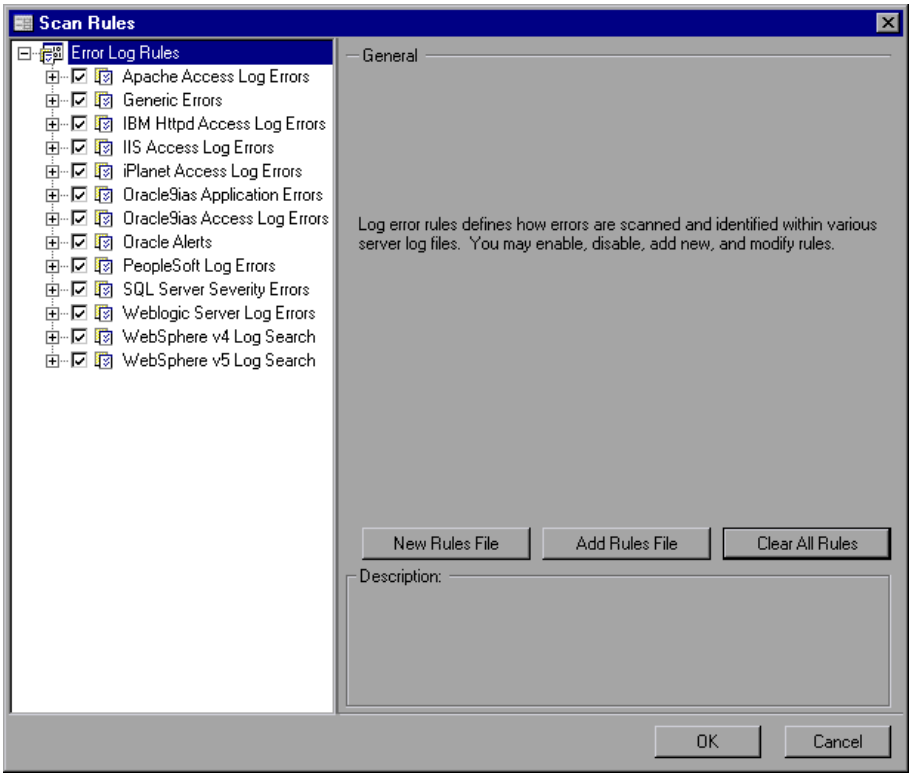
6 Use the up and down arrows to move to the next or previous errors.

To stop the scan:

➤ Click **Stop Scan**.

To specify how the Console searches for errors in the log files:

1 Click **Scan Rules**. The Scan Rules dialog box opens.



2 The left pane of the dialog box displays a tree of rules that define how errors are scanned and identified in the log files. Select or clear a checkbox to enable or disable a rule, respectively.

3 To create a new rules file, click **New Rules File**. Enter a name and description of the new rules file.

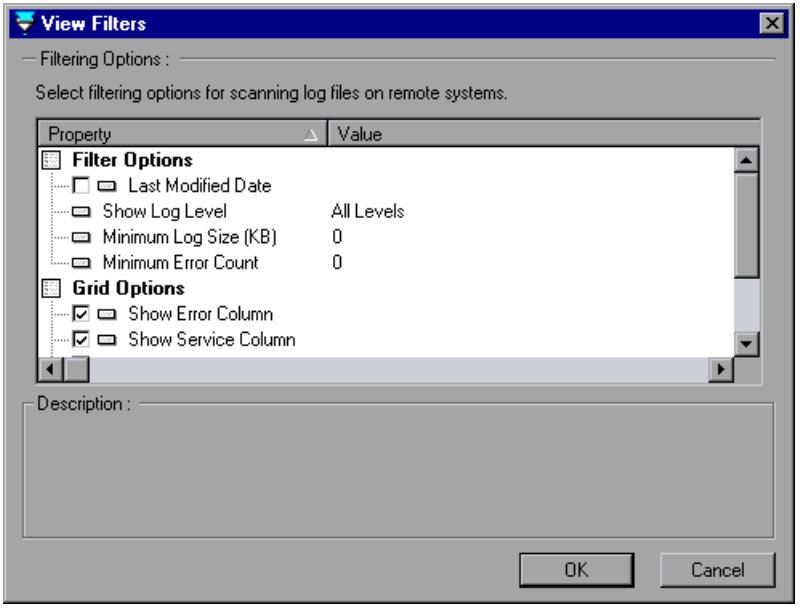
To import a rules file, click **Add Rules File** and choose the file you want to import.

To delete all rules from the rules repository, click **Clear All Rules**. Note that this operation cannot be undone.

4 Click **OK**.

To specify the way the log data is displayed:

1 Click **View Filters**. The View Filters dialog box opens.



2 In the Property column, check the items that you want to display, and clear those that you don't want. In the Value column, click a field to edit it, and enter the new value.

3 Click **OK**.

Understanding the Scan Logs Window

The Scan Logs window lets you search log files for error messages.

The upper section of the window contains the following:

Host: Allows you to choose the host machine that you want to scan.

Service: Allows you to choose the service whose log files you want to scan for errors.

Start Scan: Starts scanning all the log files used by the products defined for the topology elements in the Topology window. For example, if a component is defined with the Apache Web server, LoadRunner Tuning Module scans the Apache log file on the Web server. When the scanning is complete, LoadRunner Tuning Module displays the information about each detected log file in the table.

Scan for errors in each log file: Causes the Console to search the log files for errors when you click **Start Scan**.

Stop Scan: Stops the scanning process.

Scan Rules: Specifies how the Console searches for errors in the log files.

View Filters: Specifies the way log data is displayed.

The middle section of the window contains a table of the log file and error message information. The table is divided into the following columns:

Name: Name of the log file.

Error: Number of error messages detected in the log file.

Service: The service to which the log file belongs.

Host: Host name of the machine where the log file is located.

Size: Size of the log file, in Kilobytes.

Last Modified: When the log file was last modified.

Path: The path to the log file.

The lower part of the window contains the following:

View Log: Opens the Log Viewer.

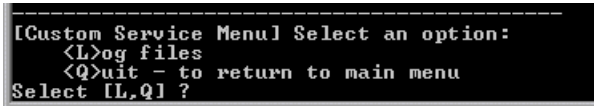
Description: A description of the selected log file, or the text of the selected error message.

Scanning Custom Service Log Files

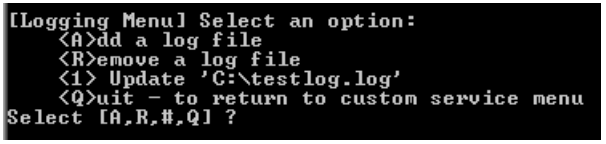
You use the pe_registry utility to define an application as a custom service and specify the location of the log file.

To define a custom service and specify the log file location:

1. Invoke the pe_registry utility (see "Using the Performance Tuner Registry" on page 704).
2. Type "L" and press <Enter> to list the current tuners.
3. Choose **Custom Service** and press <Enter>. The Custom Service menu opens.



4. Type "L" and press <Enter>. The Logging menu opens.



5. Type "A" and press <Enter>.
6. Enter the full path and filename of the log file.
7. When you are prompted to enter the user-defined file type, enter a keyword (for example, "custom"). This keyword should be used when defining error scan rules, to clearly identify files that should be scanned for specific error conditions.
8. Enter a description of the log file.
9. Press "Q" to exit the Logging menu.
10. Press "Q" to exit the Custom Service menu.

When you subsequently perform a Scan Logs step, the log file is listed and can be viewed.

Scan Recommended Values Step

The Scan Recommended Values step compares the values of parameters on a machine or service with the values recommended by LoadRunner Tuning Module's knowledge base. The step shows you the parameters that have

different values, and allows you to save Recommended Value Reports showing the differences.

Scanning for Differences in Recommended Values

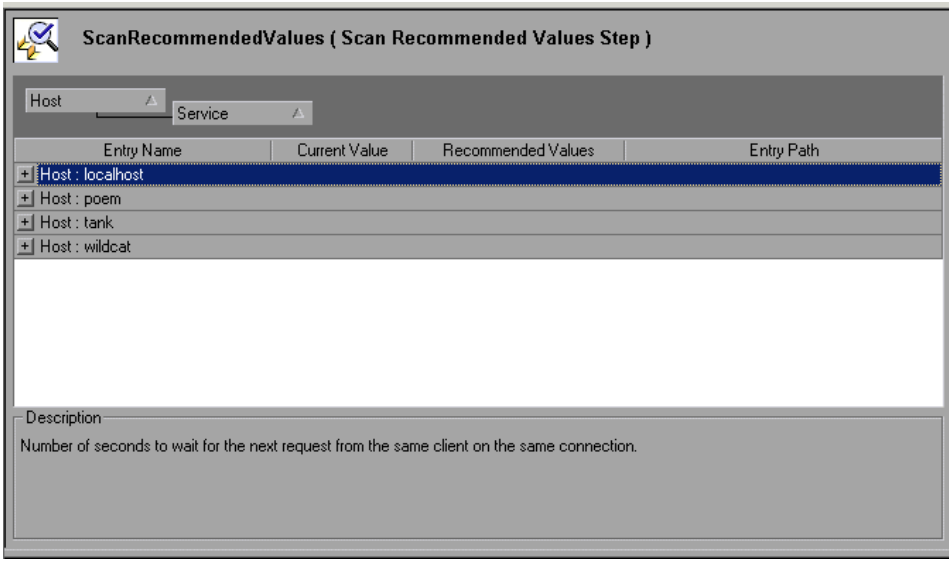
When you create a Scan Recommended Values step, LoadRunner Tuning Module displays a table showing the current and recommended values of the services running on the host machines, if there are any differences. If there are no differences between a host's current values and the recommended values for its services, the host does not appear in the table.

Note: The table contains data only after you perform a Scan Recommended Values step. When you initially create or display a Scan Recommended Values step, the table is empty.

To view the differences in recommended values:

1 Make sure that tuning agents are running on the host machines, and then click **Scan Now**. The Console displays a report showing the services containing values that differ from the recommended ones.

You can group the values as needed by dragging the column names. In the following example, the window displays the services in each host.



2 To view the services and their differing values, expand the relevant elements.

Entry Name	Current Value	Recommended Values	Entry Path
Host : localhost			
Service : Apache Web Server 1.x/2.x			
Keep-Alive Timeout	30	60	Apache Web Server 1.x/2.x / Apache Server
Send/Recv Timeout	11	120	Apache Web Server 1.x/2.x / Apache Server
Threads Per Child	22	200	Apache Web Server 1.x/2.x / Apache Server
Keep-Alive Timeout	15	60	Apache Web Server 1.x/2.x / Apache Server
Send/Recv Timeout	88	120	Apache Web Server 1.x/2.x / Apache Server
Threads Per Child	33	200	Apache Web Server 1.x/2.x / Apache Server
Service : Operating System			
Host : poem			
Host : tank			
Host : wildcat			

3 When you click a property in the table, the Console displays detailed recommendations for the property in the Description field.

Saving Recommended Values Reports

To save a report of the properties whose values differ from the recommendations:

1. Click the **Save to File** button on the toolbar. The Save As window opens.
2. Browse to the desired location, specify a filename with either an .htm or .xls extension, and click **Save**.

   LoadRunner Tuning Module saves a report of the recommended values in a file with the name you specified.

21

Generating a Session Report

LoadRunner Tuning Module allows you to generate a report of the activity that took place during a session. The report is in Microsoft Word format.

This chapter describes:

- Generating the Report
- Viewing Step Run Information
- Viewing Tuning Information
- Adding Session Notes
- Deleting Session Report Entries

About Generating a Session Report

The Session Report includes:

- information relating to the entire session, including the user-specified objectives and the system topology
- step information

Since a LoadRunner Tuning Module session can contain multiple steps, the report includes a separate section for each step. The section includes the following subsections:

- Objectives
- Conclusions
- Execution Notes
- Step Summary

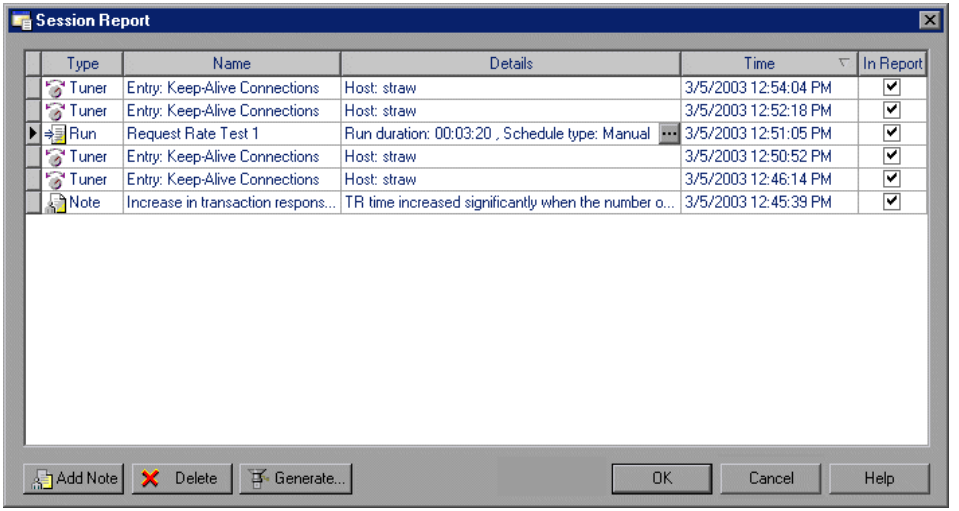
- Scheduler Information
- Scripts
- Run Time Settings
- Transaction Response Times
- Transaction Response Times Under Load
- Hardware Utilization - Windows Resources

Generating the Report

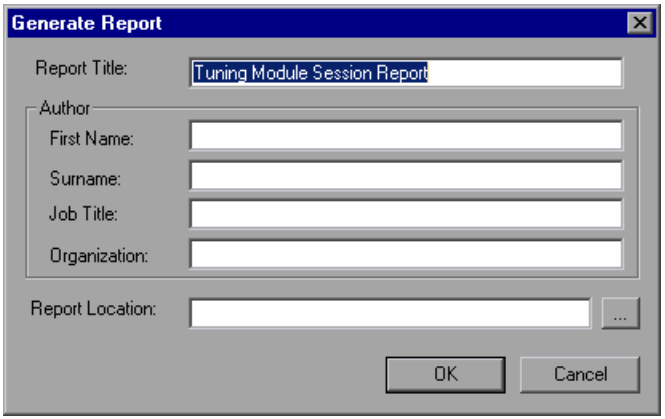
After you've executed all the steps that you want to include in the session, you generate the Session Report as follows:

1 Click **Report**, or choose **Session** > **Session Report**.

The Session Report dialog box appears.



2 Click **Generate**. The Generate Report dialog box is displayed..



3 Enter information in the fields and click **OK**. LoadRunner Tuning Module generates the report in Word format.

To view the report, locate it in the directory that you specified in the Report Location field, and open it with Microsoft Word.

Understanding the Session Report Dialog Box

The Session Report dialog box shows the activity that took place during a session, and lets you specify which information should appear in the session report.

Type: One of the following:

- **Run**—A step that was executed
- **Tuner**—A tuning action
- **Note**—A note entered by the user

Name: The result name, name of the tuned (changed) property, or subject of the note.

Details: Duration of the step run and the schedule type—Manual or Goal-Oriented (for a step run), host name or IP address of the tuned host (for a tuner entry), or the subject of the note. Click the Browse button for more details.

Time: When the step run began, the property was tuned, or the note was created.

In Report: Checking this box causes the entry to be included in the session report.

Add Note: Adds a session note to the Session Report. For more information, see "Adding Session Notes," on page 245.

Delete: Deletes the selected entry from the Session Report.

Understanding the Generate Report Dialog Box

The Generate Report dialog box is used for specifying various types of information that should appear in the generated session report.

Report Title: Enter a name for the report, or use the default name.

First Name, **Surname**, **Job Title** and **Organization:** Optional fields.

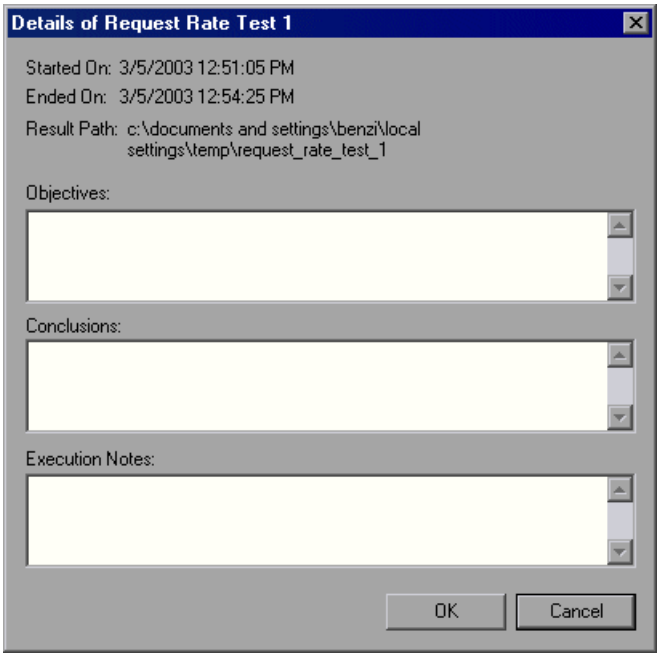
Report Location: Click the **Browse** button to specify the report's directory and filename.

Viewing Step Run Information

To view additional information about a step's execution:

1 In the Session Report dialog box, click the step's row to select it.

2 Click the Browse button in the Details column. The Step Details dialog box is displayed.



3 Make any necessary changes and click **OK** to save your changes, or **Cancel** to exit without saving.

Understanding the Step Details Dialog Box

The Step Details dialog box contains the following information:

- **Started On**—When the step started executing
- **Ended On**—When the step execution ended
- **Result path**—Path to the directory where the step's results are stored

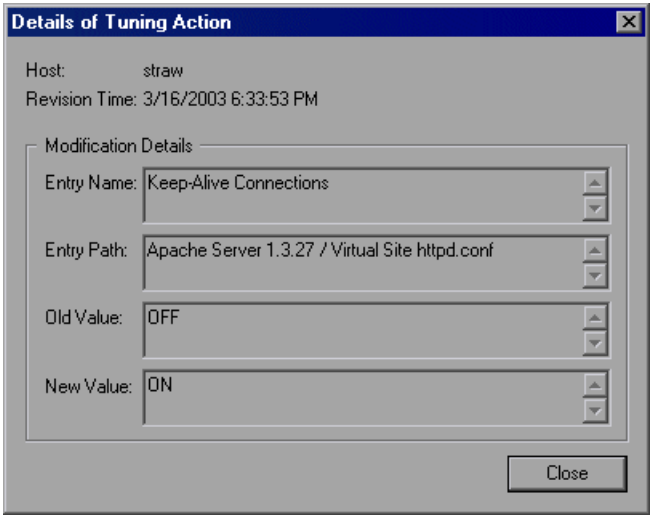
➤ **Objectives, Conclusions and Execution Notes**: You use these fields to enter relevant information about the step's execution.

Viewing Tuning Information

To view additional information about a tuning action:

1 In the Session Report dialog box, click the tuning action's row to select it.

2 Click the row's Browse button. The Tuning Action Details window is displayed.



3 Click **Close** to exit the window.

Understanding the Tuning Action Details Window

The Tuning Action Details Dialog Box displays the following information about the tuning action:

Host Name—Host name of the tuned machine.

Revision Time—When the property was tuned.

Entry Name—Name of the tuned (changed) property

Entry Path—Logical path to the entry (as seen in the Server Configurations tree in the Configuration tab)

Old Value—The property's value prior to the change

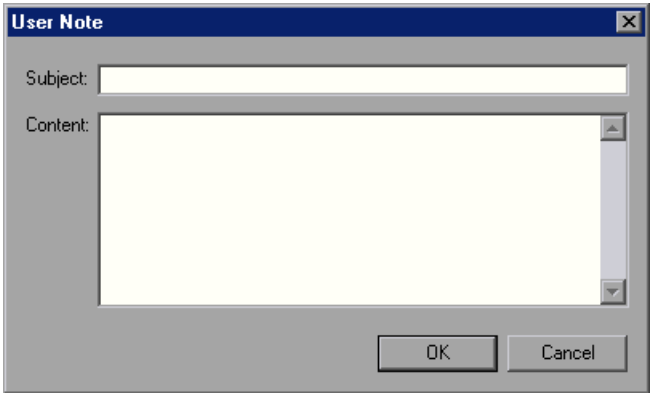
New Value—The property's value after the change

Adding Session Notes

You can add session notes in the Session Report dialog box. These notes can then be included in the Session Report.

To add a session note:

1 Click **Add Note**. The User Note dialog box appears.



2 Enter the subject and content of the note in the relevant fields.

3 Click **OK** to save the note. The note appears as an entry in the Session Report dialog box.

To subsequently view the note, select the entry and click the Browse button. LoadRunner Tuning Module displays the note, along with its creation and modification dates.

Deleting Session Report Entries

To delete an entry in the Session Report dialog box:

➤ Select the entry and click **Delete**.

Part IV

Working with Firewalls

22

Using Firewalls in LoadRunner Tuning Module

You can run Vusers and monitor your servers within a firewall, while the Console is outside of the firewall.

This chapter describes:

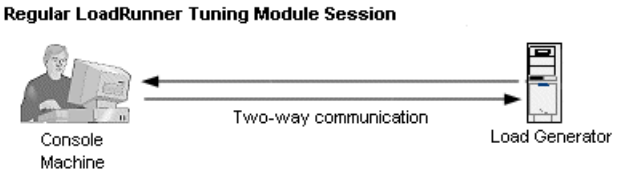
- About Using Firewalls in LoadRunner Tuning Module
- Configuring Your System
- Overview of Running or Monitoring Vusers over a Firewall
- Checking Connectivity

About Using Firewalls in LoadRunner Tuning Module

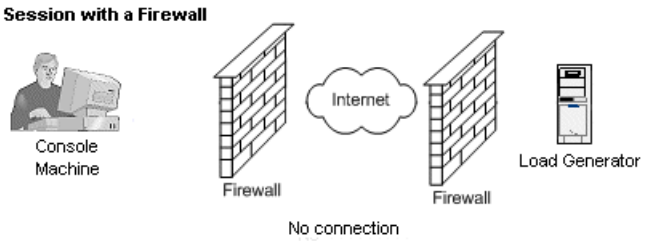
Working with a firewall means that you can prevent unauthorized access to or from a private network, on specific port numbers.

For example, you can specify that there is no access to any port from the outside world, with the exception of the mail port (23), or you can specify that there is no outside connection to any ports except for the mail port and WEB port (80). The port settings are configured by the system administrator.

In a regular LoadRunner Tuning Module session (not over the firewall), the Console has direct access to the LoadRunner agents running on remote machines. This enables the Console to connect directly to those machines.



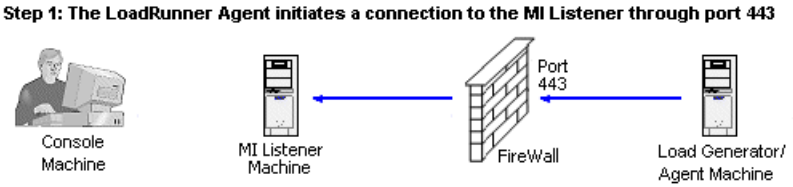
When running Vusers or monitoring servers over the firewall, this direct connection is blocked by the firewall. The connection cannot be established by the Console, because it does not have permissions to make an opening in the firewall.



LoadRunner Tuning Module solves this problem by using a communication mechanism based on HTTPS or secured TCP/IP that uses the standard SSL port on the firewall (port 443). For more information on system configuration, see "Configuring Your System" on page 252.

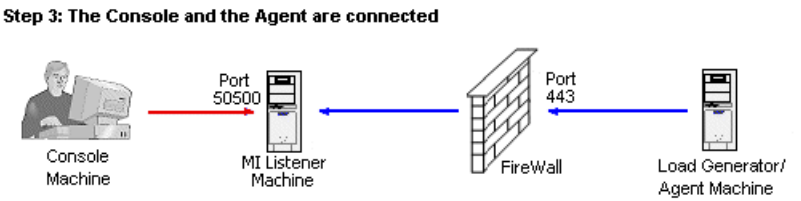
A LoadRunner agent is installed inside the firewall on either load generator machines running Vusers, or on agent machines acting as mediators for the servers to be monitored (referred to as "mediators"). The agent communicates with the Mercury Interactive listener machine, MI Listener, through port 443 in the firewall.

The MI Listener is a component that serves as router between the Console and the LoadRunner agent.



When the LoadRunner agent makes a connection to the MI Listener, the MI Listener keeps a listing of the connection to the agent using a symbolic name that the agent passed to it. When the Console connects to the MI Listener, it communicates to the MI Listener through port 50500.

The Console uses a symbolic name for the agent, and gives the MI Listener machine's name. If there has been a connection from the agent with the same symbolic name to this MI Listener, the connection is made. You can now run or monitor Vusers over a firewall.

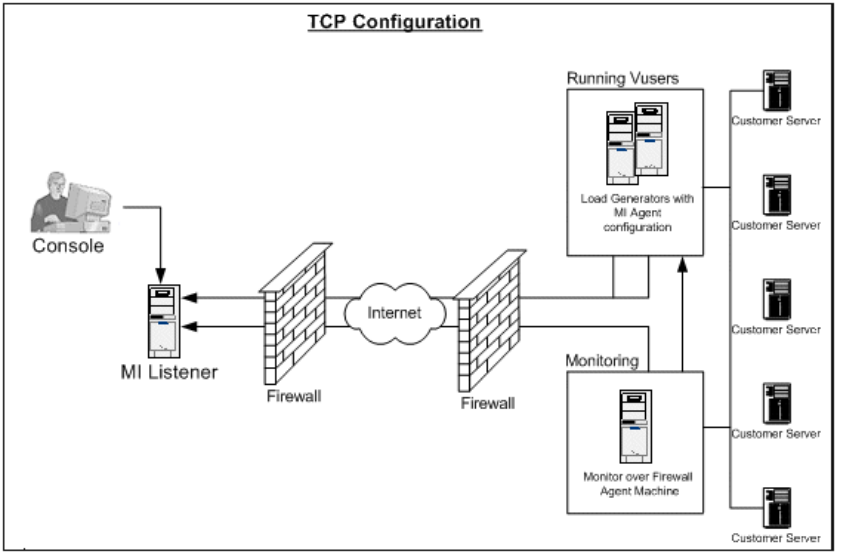


Configuring Your System

To run Vusers or monitor servers over the firewall, configure your system according to the HTTPS or secured TCP/IP configuration. Note that these configurations contain a firewall on each LAN. There may also be configurations where there is a firewall only for LAN1.

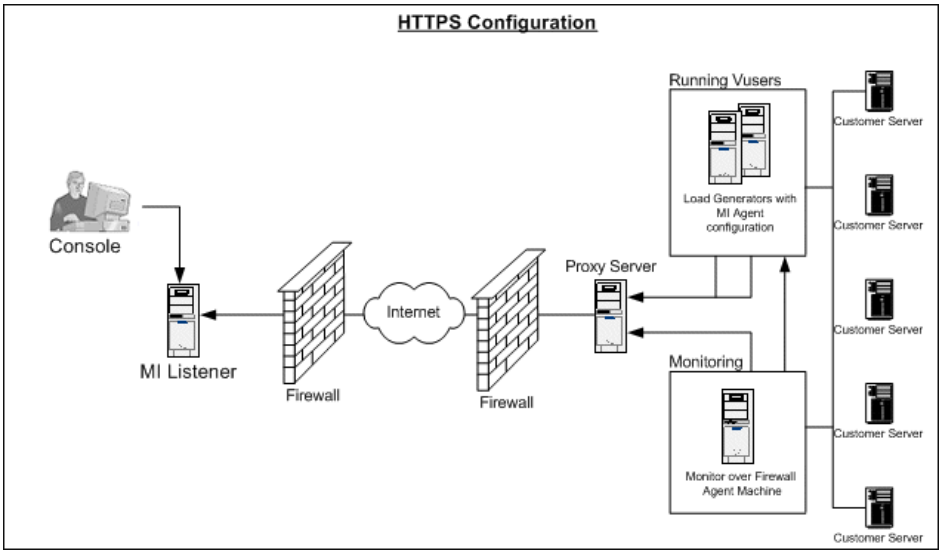
TCP Configuration

The TCP configuration requires every LoadRunner agent machine behind the firewall to be allowed to open a port in the firewall for outgoing communication.



HTTPS Configuration

In the HTTPS configuration, only one machine (the proxy server) is allowed to open a port in the firewall. Therefore it is necessary to route all outgoing communications through the proxy server.



Overview of Running or Monitoring Vusers over a Firewall

The ability to run Vusers and monitor servers over a firewall is critical for successful load testing. To prepare the LoadRunner Tuning Module for running Vusers or monitoring servers over the firewall, perform the relevant installation, configuration, and connection procedures. These procedures are explained in detail in Chapter 23, "Running Vusers Over a Firewall,"and Chapter 24, "Monitoring Over a Firewall." Note that procedures 4 and 8 are required only if you are monitoring over a firewall.

Note: The LoadRunner Agent and MI Listener installation and configuration procedures are the same for both LoadRunner and the LoadRunner Tuning Module. If you have already completed these procedures for LoadRunner, you only need to configure the Console machine to recognize the agent and MI Listener machines.

---

1 **Install the LoadRunner agent on the machines running Vusers, or on the servers to be monitored inside the firewall.**

Check that the agent is installed on the machines running Vusers, or on the servers to be monitored inside the firewall. LoadRunner agents can run on Windows or Unix machines. See "Installing LoadRunner Agents Inside the Firewall" on page 261.

2 **Configure the LoadRunner agent to operate over the firewall.**

Configure the LoadRunner agent on the machines running Vusers, or acting as mediators for the servers to be monitored. See "Configuring LoadRunner Agents Inside the Firewall" on page 262 for instructions.

3 **Configure the firewall(s).**

Configure the firewall to allow communication between the agents inside the firewall, and the machines outside the firewall. See "Configuring the Firewall to Allow Agent Access" on page 271.

4 **Install the Monitoring over Firewall Component (monitoring over a firewall only).**

To monitor a server over the firewall, install this component on the agent machine that sits inside the firewall. This machine acts as a mediator between the Console and the monitored server. See the diagrams in "Configuring Your System" on page 252 for information about where to install the Monitoring over the Firewall component. For configuration instructions, see "Installing Monitors over Firewall" on page 278.

5 **Install the MI Listener on a machine outside the firewall.**

For information about installing the MI Listener, refer to the *LoadRunner Tuning Module Installation Guide*. See the diagrams in "Configuring Your System" on page 252 for information about where to install the MI Listener.

6 **Configure the MI Listener machines.**

Configure the security attributes on each MI Listener machine. See "Installing and Configuring the MI Listener Outside the Firewall" on page 272.

7 **Configure the Console machine.**

Configure the Console machine to recognize the agent and MI Listener machines. See "Configuring the Console to Run or Monitor Vusers Over a Firewall" on page 274.

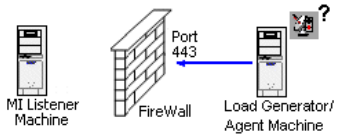
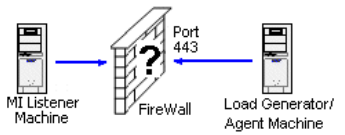
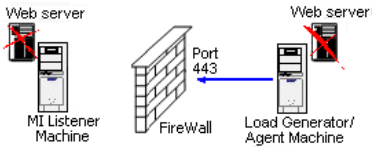
8 **Configure Server Monitor Properties (monitoring over a firewall only).**

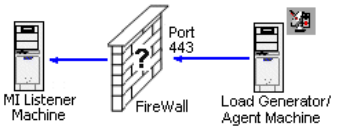
Configure the Server Monitor Properties and measurement frequency. See "Configuring Server Monitor Properties" on page 279, "Adding and Removing Measurements" on page 282, and "Configuring Measurement Frequency" on page 283.

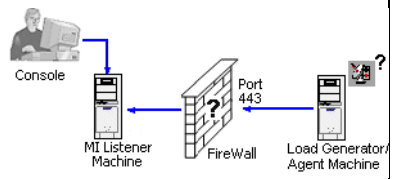
Checking Connectivity

To run Vusers or monitor servers over a firewall, you must be able to establish a connection between the LoadRunner agent, MI Listener, and the Console machine.

If you encounter connectivity problems after installing and configuring all the necessary components, check the table below for troubleshooting tips.

Check	Solution
To check that the Firewall service was activated on the agent machine:  	There should be a traffic light on the right side of the LoadRunner Agent icon on the machine running/ monitoring Vusers over a firewall. If there is no traffic light, this indicates that the 'FirewallServiceActive=1' is not set in the [FireWall] section of the Agent Settings. See "Configuring and Running the Windows LoadRunner Agent" on page 262.
To check that port 443 is open:  	On the agent machine, open a Command Prompt window, and type the following: telnet <MI_Listener_IP>443. For example: telnet 111.111.111.1111 443. If port 443 is open, a new Telnet window will open. If port 443 is not open, contact your network administrator. Note: Running Vusers over a firewall requires bi-directional communications. Therefore, you must also run this test from the MI Listener. Type: telnet <agent_IP>443.
To check that port 443 is available:  	If a Web server is running on the MI Listener or Monitor over Firewall machine, port 443 will not allow the access required by the listening and monitoring processes. Contact your network administrator to change the Web server port.

Check	Solution
To check connectivity between the agent and the MI Listener, when running the LoadRunner agent as a service:  	If there is a red light on the right side of the LoadRunner Agent icon when running the LoadRunner agent as a service, do the following: • Check that port 443 is open. See the troubleshooting tip above. • Check that the Agent Settings and Agent Configuration are correctly set. See "Configuring and Running the Windows LoadRunner Agent" on page 262. • Run the agent as a Process. Launch **<LoadRunner Tuning Module Installation>\Launch_service\bin\magentproc.exe**. If this works, this indicates an authentication issue with the LoadRunner Tuning Module Agent Service. Browse to the Service > LoadRunner Tuning Module Agent Service, and change the properties of this service to 'System User Account' or provide the username and password of someone who has administrative privileges on this machine.

Check	Solution
To check connectivity between the agent and the Console, when monitoring over a firewall  	• Check that you entered the servers that you want to monitor in the Monitor Configuration dialog box. (See "Configuring Server Monitor Properties" on page 279). • Start the LoadRunner Tuning Module Agent Process on the mediator machine. (See "Configuring LoadRunner Agents Inside the Firewall" on page 262). • On the Console, enter the name of the mediator machine in the Load Generators dialog box, and click **Connect**. After about a minute, data should start streaming in from the mediator through the MI Listener to the Console. (See "Configuring the Console to Run or Monitor Vusers Over a Firewall" on page 274). • If no data arrives at the Console, try connecting the Console to the MI Listener as if the Listener were used as a load generator. This will help identify the cause of the problem. Examine the log file on the mediator machine by right-clicking the LoadRunner Agent icon. There should be no error messages. • Start the MI Listener, and then manually start the LoadRunner Agent Process by running **<LoadRunner Tuning Module installation>\launch_service\bin\magentproc.exe** on the mediator machine. Allow the mediator machine sufficient time to connect to the MI Listener, then connect the Console to the mediator machine. If the LoadRunner Agent Process crashes, either restart the agent or reboot the mediator machine.

23

Running Vusers Over a Firewall

The LoadRunner Tuning Module enables you to run Vusers over a firewall, while the Console is outside of the firewall.

This chapter describes:

- About Running Vusers Over a Firewall
- Overview of Running Vusers Over a Firewall
- Installing LoadRunner Agents Inside the Firewall
- Configuring LoadRunner Agents Inside the Firewall
- Configuring the Firewall to Allow Agent Access
- Installing and Configuring the MI Listener Outside the Firewall
- Configuring the Console to Run or Monitor Vusers Over a Firewall

About Running Vusers Over a Firewall

To prepare the LoadRunner Tuning Module for running Vusers over the firewall, perform all the procedures in "Overview of Running Vusers Over a Firewall" on page 260. To enable monitoring of your servers from outside the firewall, you must also complete the procedures in "Monitoring Over a Firewall" on page 277.

For a complete overview of the installation, configuration, and connection requirements when working with firewalls, see "Overview of Running or Monitoring Vusers over a Firewall" on page 254.

Overview of Running Vusers Over a Firewall

The LoadRunner Agent and MI Listener installation and configuration procedures are the same for both LoadRunner and the LoadRunner Tuning Module. If you have already completed these procedures for LoadRunner, you only need to configure the Console machine to recognize the agent and MI Listener machines.

1 **Install the LoadRunner agent on the machines running Vusers inside the firewall.**

See "Installing LoadRunner Agents Inside the Firewall" on page 261.

2 **Configure the LoadRunner agent to operate over the firewall.**

See "Configuring LoadRunner Agents Inside the Firewall" on page 262 for instructions.

3 **Configure the firewall(s) to allow communication between the agents inside the firewall, and the machines outside the firewall.**

See "Configuring the Firewall to Allow Agent Access" on page 271.

4 **Install the MI Listener on a machine outside the firewall.**

For installation information, refer to the *LoadRunner Tuning Module Installation Guide*.

5 **Configure the security attributes on each MI Listener machine.**

See "Installing and Configuring the MI Listener Outside the Firewall" on page 272.

6 **Configure the Console machine to recognize the agent and MI Listener machines.**

See "Configuring the Console to Run or Monitor Vusers Over a Firewall" on page 274.

Installing LoadRunner Agents Inside the Firewall

To run or monitor Vusers over a firewall, a LoadRunner agent must be installed on the load generator machines running Vusers, or on the servers to be monitored inside the firewall. The agent is added either as a Windows service or as an executable run from the Startup folder.

Running Vusers over a Firewall

A LoadRunner agent may already be installed on load generator machines inside the firewall if you ran the installation from setup. Click **Start** > **Programs** > LoadRunner Tuning Module > **Tuning Module Agent Service/Process** to check whether it is installed. If **Agent Service** or **Agent Process** appears on the list of LoadRunner Tuning Module options, then the agent was already installed.

If there is no agent installed, install the Load Generator component from the LoadRunner Tuning Module Console CD on the machine(s) running Vusers inside the firewall. See the diagrams in "Configuring Your System" on page 252 for information about where to install the Load Generator component.

Monitoring over a Firewall

Install the Monitors over Firewall component on the servers to be monitored inside the firewall. For more information, see "Installing Monitors over Firewall" on page 278. See the diagrams in "Configuring Your System" on page 252 for information about where to install the Monitors over the Firewall component.

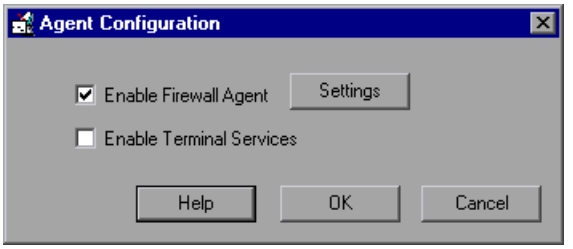
Configuring LoadRunner Agents Inside the Firewall

The machines inside the firewall can either be Load Generator machines running Vusers, or mediator machines connected to the servers to be monitored by the Console. You configure the LoadRunner agents inside the firewall to operate over the firewall. The Console machine resides outside the firewall.

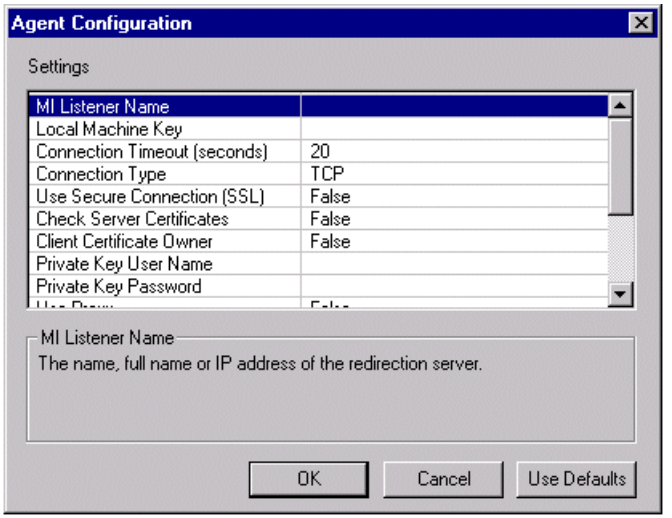
Configuring and Running the Windows LoadRunner Agent

To configure the LoadRunner agent on Windows machines:

1. Stop the LoadRunner agent by right-clicking its icon in the system tray and selecting **Close**.
2. Run **Agent Configuration** from **Start** > **Programs** > LoadRunner Tuning Module > **Advanced Settings**, or run <LoadRunner Tuning Module **root**>**\launch_service\bin\AgentConfig.exe**.
3. Select the **Enable** check box, and then click **Settings**.



The Agent Configuration dialog box opens.



4 Set each option as described in "Agent Configuration Settings" on page 269.

5 Click **OK** to save your changes, **Cancel** to cancel them, or **Use Defaults**.

6 Restart the LoadRunner agent by double-clicking the shortcut on the desktop, or from **Start** > **Programs** > LoadRunner Tuning Module > **Tuning Module Agent Service/Process**.

7 Check the connection status between the LoadRunner agent and the MI Listener. A green light illuminated next to the LoadRunner Agent icon in the system tray indicates a successful connection between the LoadRunner agent and the MI Listener. A red light indicates that there is no connection between the agent and the MI Listener.

Configuring and Running the UNIX LoadRunner Agent

To configure the LoadRunner agent on UNIX machines:

1. Open <LoadRunner Tuning Module **root folder**>**/dat/br_lnch_server.cfg** in a text editor.
2. In the Firewall section, set FireWallServiceActive to 1 and save your changes.
3. Run **agent_config** from the <LoadRunner Tuning Module **root folder**>**/bin** directory to display the following menu:

```
Menu:
1. Show current settings.
2. Change a setting.
3. Save changes and exit.
4. Exit without saving.
5. Use default values.
```

4. Enter 1 to display the current settings:

```
Settings:
---------
1. MI Listener Name =
2. Local Machine Key =
3. Connection Timeout (seconds) = 20
4. Connection Type = TCP
5. Use Secure Connection (SSL) = False
6. Check Server Certificates = False
7. Client Certificate Owner = False
8. Private Key User Name =
9. Private Key Password =
10. Proxy Name =
11. Proxy Port =
12. Proxy User Name =
13. Proxy Password =
14. Proxy Domain =


Menu:
1. Show current settings.
2. Change a setting.
3. Save changes and exit.
4. Exit without saving.
5. Use default values.
```

5 To change a setting, enter 2 to display the settings menu:

```
Settings:
---------
1. MI Listener Name =
2. Local Machine Key =
3. Connection Timeout (seconds) = 20
4. Connection Type = TCP
5. Use Secure Connection (SSL) = False
6. Check Server Certificates = False
7. Client Certificate Owner = False
8. Private Key User Name =
9. Private Key Password =
10. Proxy Name =
11. Proxy Port =
12. Proxy User Name =
13. Proxy Password =
14. Proxy Domain =

Enter number of setting to change or 0 to go back to menu.
```

Enter the setting and continue according to the menu instructions. Set each option according to the "Agent Configuration Settings" on page 269.

Examples of Changing Agent Settings in Unix

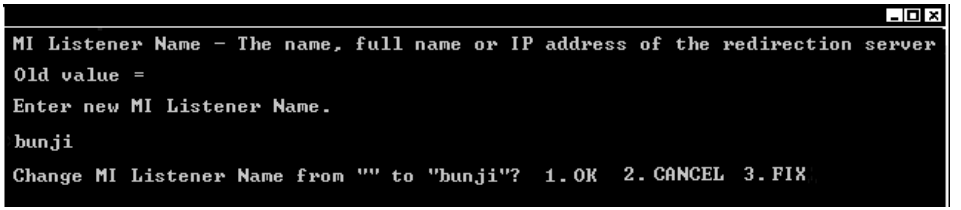
To change the MI Listener Name:

1 Enter 1 in the Settings menu to display the following screen:

```
MI Listener Name - The name, full name or IP address of the redirection server
Old value =
Enter new MI Listener Name.
```

Line 1 is a description of the setting. Line 2 shows the current value of the setting.

2 Enter the new value, (for example, 'bunji') to display the following:



3 To keep the new value and return to the menu, enter 1.

To discard the new value and return to the menu, enter 2.

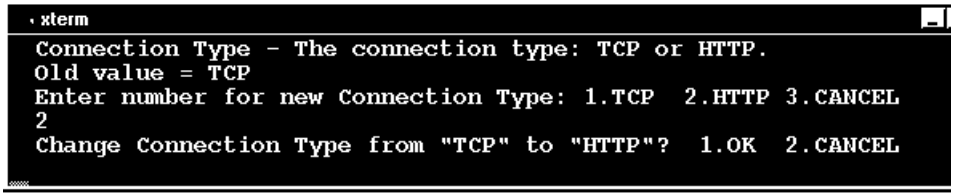
To discard the new value and change the setting once more, enter 3.

To change the Connection Type:

1 Enter 4 in the Settings menu to display the following screen:

Line 1 is a description of the setting. Line 2 shows the current value of the setting.

2 Enter 1 to set the connection type to TCP, or enter 2 to set it to HTTP and display the following:



3 To keep the new value and return to the menu, enter 1.

To discard the new value and return to the menu, enter 2.

Viewing the Settings and Restarting the Agent

To view the current settings:

1 Return to the main menu by entering 1.

2 Enter 1 to display the settings. The following example includes the new settings for MI Listener Name and Connection Type:

3 To save your changes, enter 3 from the main menu.

To cancel your changes, enter 4.

To use the default values supplied by LoadRunner Tuning Module (as described in "Agent Configuration Settings" on page 269), enter 5.

To start or remove the LoadRunner agent:

1. To start the LoadRunner agent, run the command m_daemon_setup -install from the <**LoadRunner Tuning Module root folder**>/**bin** directory.
2. To remove the LoadRunner agent, run the command m_daemon_setup -remove from the <**LoadRunner Tuning Module root folder**>/**bin** directory.

Note: When the LoadRunner agent is configured to run over a firewall, and the agent is connected to the MI Listener, a file called <local_machine_key>_connected_to_MI_Listener is created in the temporary directory of the LoadRunner agent machine. The file is removed when the LoadRunner agent disconnects from the MI Listener.

For more information about running the LoadRunner agent, refer to "UNIX Shell" in Appendix B, "Troubleshooting the Console."

Agent Configuration Settings

Option	Default Value	Description
MI Listener name	none	The name, full name, or IP address of the Mercury Interactive listener machine, MI Listener.
Local Machine Key	none	A symbolic string identifier used to establish a unique connection between the Console host behind the firewall, and the agent machine (via the MI Listener machine).
Connection Timeout (seconds)	20 seconds	The length of time you want the agent to wait before retrying to connect to the MI Listener machine. If zero, the connection is kept open from the time the agent is run.
Connection Type	TCP	Choose either TCP or HTTP, depending on the configuration you are using.
Server User Name	none	The username needed to connect to the MI Listener machine.
Server Password	none	The password needed to connect to the MI Listener machine.
Server Domain	none	The domain name needed to connect to the MI Listener machine. This field is only required if NTLM is used.
Use Secure Connection (SSL)	False	Choose **True** to connect using the Secure Sockets Layer protocol.

Option	Default Value	Description
Check Server Certificates	None	Authenticates the SSL certificates that are sent by the server. Choose **Medium** to verify that the server certificate is signed by a trusted Certification Authority. Choose **High** to verify that the sender IP matches the certificate information. This setting is only available if **Use Secure Connection** is set to **True**.
Client Certificate Owner	False	Choose **True** to load the SSL certificate. In some cases, the server requests a certificate to allow the connection to be made. This option is relevant only if the **Use Secure Connection** option is set to **True**.
Private Key Password	None	The password that may be required during the SSL certificate authentication process. This option is relevant only if the **Client Certificate Owner** option is set to **True**.
Proxy Name	<IE proxy server name> or None	The name of the proxy server. This option is mandatory if the **Connection Type** option is set to **HTTP.**
Proxy Port	<IE proxy server port> or None	The proxy server connection port. This option is mandatory if the **Connection Type** option is set to **HTTP**.
Proxy User Name	None	The username of a user with connection rights to the proxy server.

Option	Default Value	Description
Proxy Password	None	The user's password.
Proxy Domain	None	The user's domain if defined in the proxy server configuration. This option is required only if NTLM is used.

Configuring the Firewall to Allow Agent Access

You modify your firewall settings to enable communication between the machine(s) inside the firewall and machines outside the firewall.

TCP configuration

The LoadRunner agent tries to establish a connection with the MI Listener using port 443. To enable this connection, allow an outgoing connection for HTTPS service on the firewall for port 443. As a result, the agent will keep trying to connect to the MI Listener at an interval of the number of seconds specified in the Connection Timeout field in the agent configuration. Then the MI Listener connects back to the agent. From this point on, the agent listens to commands from the MI Listener.

HTTPS configuration

The LoadRunner agent tries to establish a connection with the MI Listener using the proxy port specified in the Proxy Port field. To enable this connection, allow an outgoing connection for HTTPS service on the firewall for port 443. The agent will keep trying to connect to the MI Listener at an interval of the number of seconds specified in the Connection Timeout field in the agent configuration. On successful connection, the agent on the proxy server connects to the MI Listener, and the MI Listener connects back to the agent through the proxy server. From this point on, the agent listens to commands from the MI Listener.

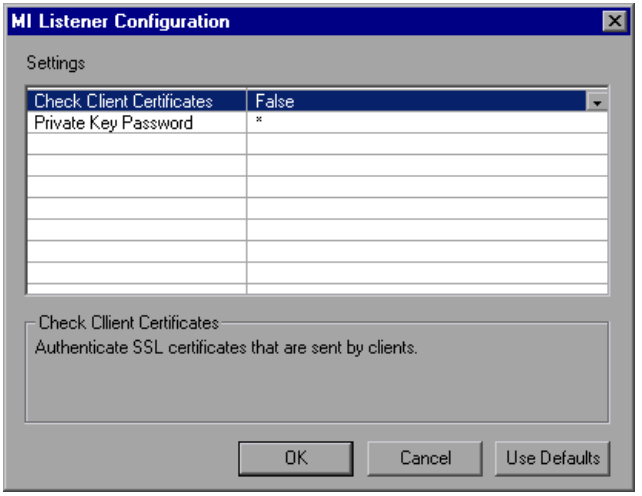
Installing and Configuring the MI Listener Outside the Firewall

To enable running Vusers or monitoring over a firewall, you need to install the MI Listener on one or more machines in the same LAN as the Console outside the firewall. For installation instructions, refer to the *LoadRunner Tuning Module Installation Guide*. Note that the Console installation automatically includes the MI Listener, so you can designate the Console as the MI Listener machine.

Note: MI Listener can only be installed on Windows machines.

To configure the MI Listener security attributes:

1. Open incoming HTTPS service for port 443. The port settings are set by your system administrator.
2. Stop the LoadRunner agent on the MI Listener machine by right-clicking its icon in the system tray and selecting **Close** from the popup menu.
3. Run **MI Listener Configuration** from **Start** > **Programs** > **LoadRunner Tuning Module** > **Advanced Settings**, or run <**LoadRunner Tuning Module_root_dir**>**launch_service****bin****MILsnConfig.exe**.



4 Set each option as described in "MI Listener Configuration Settings" on page 273.

5 Click **OK** to save your changes, **Cancel** to cancel them, or **Use Defaults**.

6 Restart the LoadRunner agent by double-clicking the shortcut on the desktop, or running it from **Start** > **Programs** > **LoadRunner Tuning Module**.

7 Make sure that port 443 is free on the MI Listener machine.

Note: Ensure that no Web Servers are running on the MI Listener or Monitor over Firewall machine. These servers use port 443 and will not allow the access required by the listening and monitoring processes.

MI Listener Configuration Settings

Option	Default Value	Description
Check Client Certificates	False	Choose **True** to request that the client send an SSL certificate when connecting, and to authenticate the certificate.
Private Key Password	None	The password that may be required during the SSL certificate authentication process.

Configuring the Console to Run or Monitor Vusers Over a Firewall

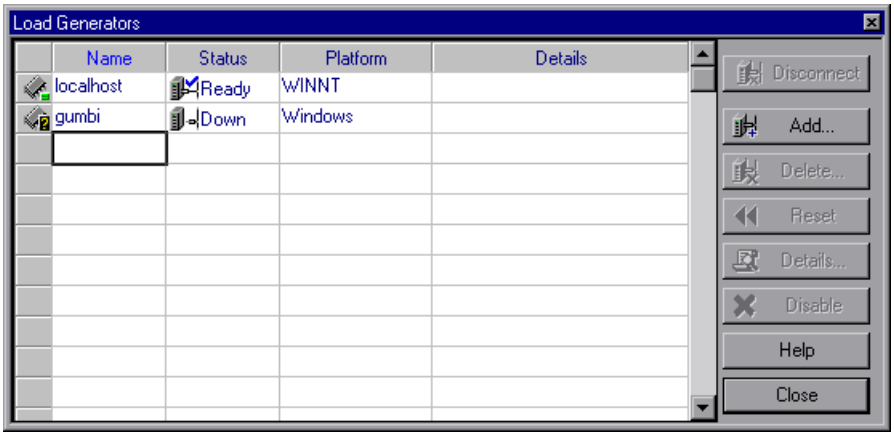
To run Vusers or monitor servers inside the firewall, you need to create a unique connection between the Console and the agent machine. This connection is made through the MI Listener, which serves as router between the Console and the LoadRunner agent. To establish this connection, you configure the Console machine to define the agent machine as a load generator.

To configure the Console for running vusers or monitoring over the firewall:

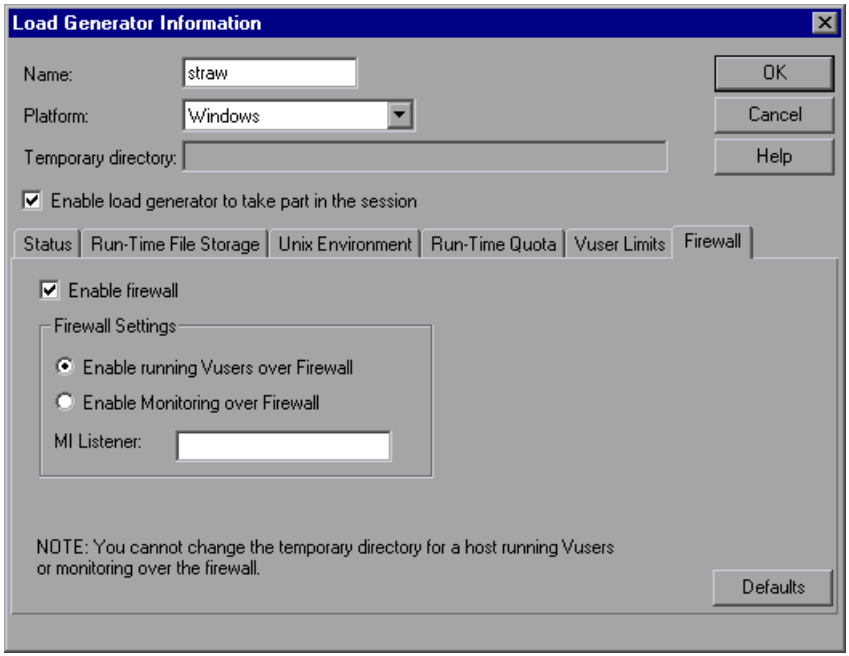
1 Run the Console from **Start** > **Programs** > **LoadRunner Tuning Module** and create a new session, or load an existing one.

2 Click **Generators** to display the Load Generators window. In the Name field, enter the symbolic name of the server. This is the same name that you entered in the Local Machine Key setting in the Agent Configuration. (See "Agent Configuration Settings" on page 269). In the example below, the server name is gumbi.

If the server is a UNIX server, change the Platform field to **UNIX**.



3 Select the Load Generator, and click **Details** to display the Load Generator Information.



4 In the Firewall tab, enter the MI Listener machine's name in the MI Listener field. This is the same name that you entered in the Agent Configuration dialog box, in the MI Listener Name setting. In this example, the MI Listener is bunji.

5 In the Firewall Settings section, choose one of the following options:

- **Enable running Vusers over Firewall**: To run Vusers over the firewall.
- **Enable Monitoring over Firewall**: To monitor Vusers over the firewall.

6 Click **OK** to return to the Load Generators dialog box.

7 Select the Load Generator and click **Connect**.

Note: Remember that you cannot change the temporary directory on the host running or monitoring Vusers over the firewall.

If you encounter connectivity problems, see "Checking Connectivity" on page 256. For other firewall troubleshooting, see "Troubleshooting Firewalls" on page 748.

24

Monitoring Over a Firewall

You can monitor your servers within a firewall while the Console is outside of the firewall.

This chapter describes:

- About Monitoring over a Firewall
- Installing Monitors over Firewall
- Preparing for Data Collection
- Configuring Server Monitor Properties
- Configuring the Network Delay Monitor over a Firewall

About Monitoring over a Firewall

To enable monitoring of your servers from outside the firewall, you must complete all the steps outlined in "Overview of Running Vusers Over a Firewall" on page 260, and perform the following procedures:

- Install the Monitoring over Firewall component
- Configure the Server Monitor Properties

The Monitors over Firewall component must be installed on designated machines inside the firewall. The installation sets up the Server Monitor mediator (referred to as the "mediator") as well as the Server Monitor configuration tool. You then configure the servers to monitor, and define the specific measurements that the LoadRunner Tuning Module mediator machine collects for each monitored server.

Note: For a complete overview of the installation, configuration, and connection requirements when working with firewalls, see "Overview of Running or Monitoring Vusers over a Firewall" on page 254.

For information on running Vusers over a firewall, see "Running Vusers Over a Firewall" on page 259.

Installing Monitors over Firewall

Monitors over Firewall may have been installed during the LoadRunner Tuning Module installation. Click **Start > Programs > LoadRunner Tuning Module > Advanced Settings** to check whether it is installed. If the **Monitor Configuration** option appears on the list of LoadRunner Tuning Module options, then Monitors over Firewall was already installed, and you can proceed to "Preparing for Data Collection" on page 279.

If there is no installation, install Monitors over Firewall on the mediator machine using one of the following:

- Perform a custom installation of LoadRunner Tuning Module from the LoadRunner Tuning Module CD, choosing only the Monitors over Firewall option.
- Obtain the **Monitors over Firewall** file from the Mercury Interactive Customer Support Web site (http://support.mercuryinteractive.com). **Monitors over Firewall** is a stand-alone downloadable installation. It comes as a self-extracting installer file.

  For instructions on performing a custom installation of LoadRunner Tuning Module, refer to the *LoadRunner Tuning Module Installation Guide*.

Preparing for Data Collection

After installing the Monitors over Firewall component, you need to prepare for data collection. Check that you have completed all the steps outlined in "Running Vusers Over a Firewall" on page 259 before continuing.

To configure the Console for data collection:

1. You should already have configured the LoadRunner agent and the Console to operate over the firewall as described in "Configuring the Console to Run or Monitor Vusers Over a Firewall" on page 274.
2. Remember that in the Firewall tab of the Load Generator Information dialog box, you should enter the IP address of the MI Listener machine, and check **Enable Monitoring over Firewall**.
3. Connect to the load generator. Select **Session** > **Load Generators**, and click **Connect**. Make sure that you obtain information for the monitors configured inside the firewall.

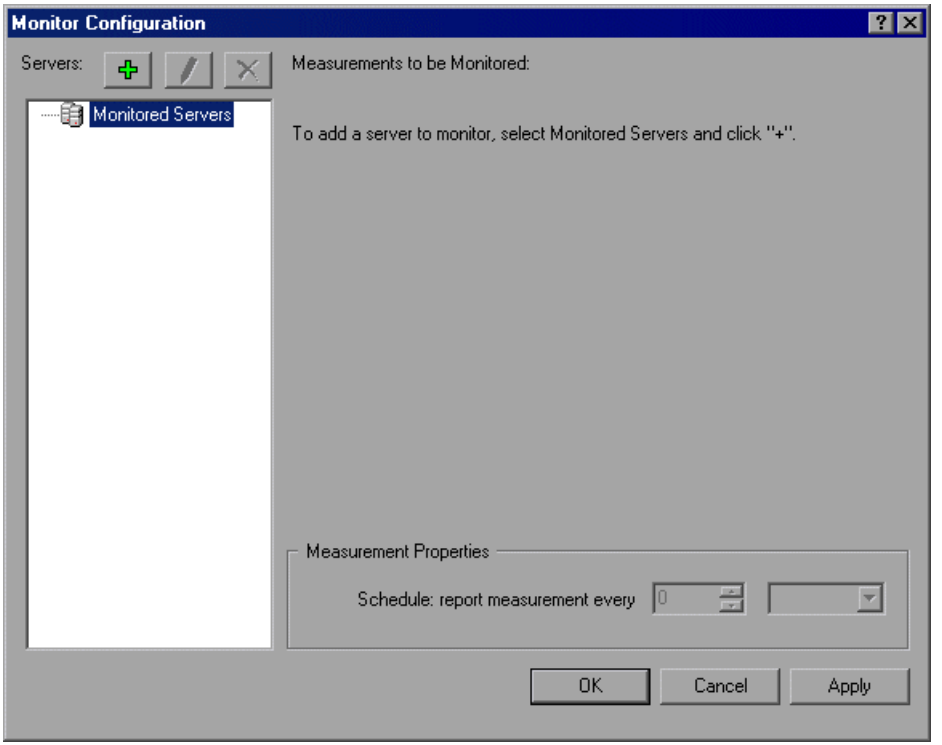
Configuring Server Monitor Properties

After you have installed and configured the LoadRunner agent, the Monitors over Firewall component, the MI Listener, and the Console machine, you need to choose the server measurements that you want the mediator machine to monitor.

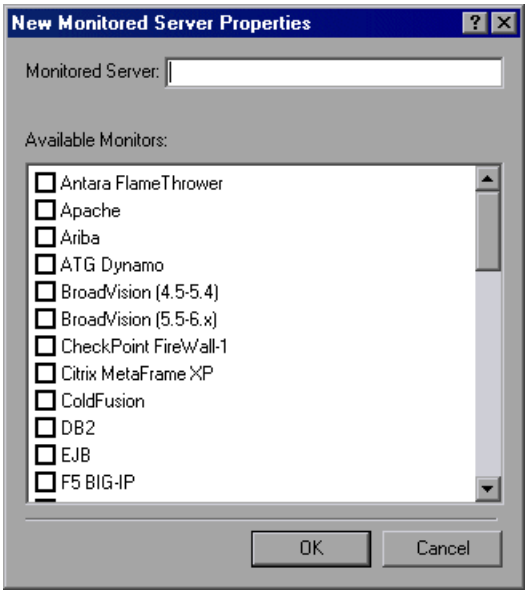
You configure the server monitor properties from the mediator machine, using the Monitor Configuration dialog box. You can select the type of monitors to run and the server whose resources you want to monitor, add the measurements to monitor for each server, and specify the frequency with which you want the monitored measurements to be reported.

To configure server monitor properties:

1 Select **Start** > **Programs** > **LoadRunner Tuning Module** > **Advanced Settings** > **Monitor Configuration**. For machines without the complete LoadRunner Tuning Module installation, select **Start** > **Programs** > **Server Monitor** > **Monitor Configuration.** The Monitor Configuration dialog box opens.



2 Click the **Add Server** button. The New Monitored Server Properties dialog box opens.



3 In the Monitored Server box, type the name or IP address of the server whose resources you want to monitor.

Note: To add several servers simultaneously, separate the server names or IP ranges with commas. For example: 255.255.255.0-255.255.255.5, server1, server2.

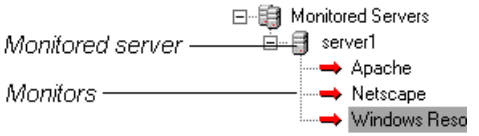
---

4 From the Available Monitors list, select the monitors appropriate for the server being monitored.

Note: Data can only be viewed for the monitors that are enabled with your LoadRunner Tuning Module license key. To preview your license key information, in the LoadRunner Tuning Module Console, select **Help** > **About LoadRunner Tuning Module**.

---

5 Click **OK** to close the New Monitored Server Properties dialog box. The Monitored Servers list is displayed in the Monitor Configuration dialog box.



Note that for certain monitors, LoadRunner Tuning Module displays default measurements in the Measurements to be Monitored section. For details on selecting measurements, see "Adding and Removing Measurements" on page 282.

6 To add additional monitored servers to the list, repeat steps 1-5.

7 To edit the monitor configuration properties for a server, click the **Edit** button. The Monitored Server Properties dialog box opens enabling you to edit the monitors for the server whose resources you are monitoring.

8 Click **Apply** to save your settings.

Adding and Removing Measurements

After you configure one or more server machines to monitor, you add measurements to monitor for each server. If LoadRunner Tuning Module added default measurements, you can edit them as required.

To add a measurement to monitor:

1 Select a server from the Monitored Servers list.

2 Click the **Add Measurement** button. Select the appropriate monitor. A dialog box opens, enabling you to choose measurements for the monitor you selected.

3 Select the measurements that you want to monitor, and click **OK**.

4 Click **Apply** to save your settings.

For information on configuring measurements for each server monitor, see the relevant chapter.

To remove a measurement from the measurements list:

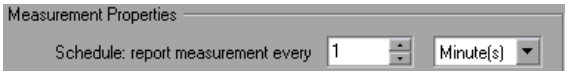
1 Select the measurement, and click the **Delete** button.

2 Click **Apply** to save your settings.

Configuring Measurement Frequency

Once you have configured monitor measurements, you configure measurement frequency.

In the Measurement Properties section, you set a measurement schedule for each measurement to be reported.



To set a measurement schedule for a measurement:

1 Select the configured server measurement you want to schedule.

2 Specify the frequency at which you want LoadRunner Tuning Module to report the measurement.

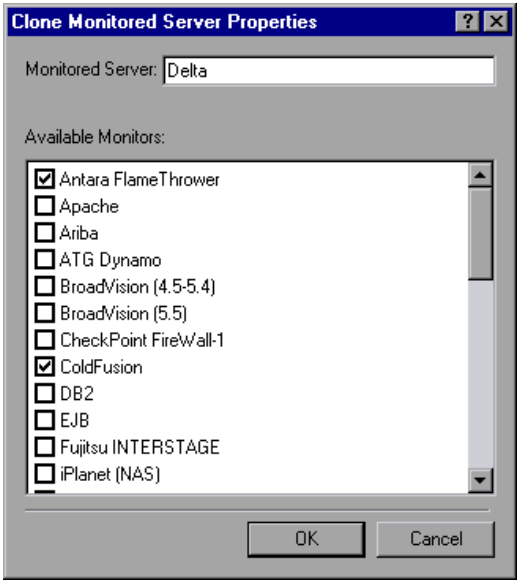
3 Click **Apply** to save your settings.

Cloning a Monitored Server's Properties

If you want to monitor the same properties on different server machines, you can clone a selected server's properties using the Clone Monitored Server Properties dialog box.

To clone a monitored server's properties:

1 In the Monitor Configuration dialog box, right-click the server you want to clone, and select **Clone**. The Clone Monitored Server Properties dialog box opens.



2 In the Monitored Server box, type the name or IP address of the clone server you want to create.

Note: To create several servers simultaneously, separate the server names or IP ranges with commas. For example: 255.255.255.0-255.255.255.5, server1, server2.

---

3. The Available Monitors list displays the monitors that were selected for the server being cloned. Select additional appropriate monitors for the clone server.
4. Click **OK** to close the Clone Monitored Server Properties dialog box. The cloned server is displayed in the Monitored Servers list.
5. Click **Apply** to save your settings.

Configuring the Network Delay Monitor over a Firewall

To run the Network Delay Monitor when there are firewalls between the Console machine and the source machine, you must configure the Network Delay Monitor (see "Configuring the Network Delay Time Monitor" on page 368), and specify the server name or IP address of the source machine in the Topology Window according to the following format:

<MI Listener machine>:<source machine local key>.

where source machine local key is the Local Machine Key that you chose when configuring the LoadRunner agent on the source machine. (See "Agent Configuration Settings" on page 269.)

For example: 12.12.12.3:vds

Part V

Monitoring a Session

25

Online Monitoring

You can monitor session step execution using the LoadRunner Tuning Module online monitors.

The specific monitors are discussed in the next few chapters. This chapter describes the online monitor user interface:

- About Online Monitoring
- Choosing Monitors and Measurements
- Viewing the Monitors
- Opening Online Monitor Graphs
- Customizing the Online Monitor Display View
- Setting Monitor Options

About Online Monitoring

LoadRunner Tuning Module enables you to view data generated during session step execution using the following online monitors:

The **Run-Time** monitor displays the number and status of Vusers participating in the session step, as well as the number and types of errors that the Vusers generate.

The **Transaction** monitor displays the transaction rate and response time during session step execution. For more information, see Chapter 27, "Run-Time and Transaction Monitoring."

The **Web Resource** monitor measures statistics at the Web server(s) during session step runs. It provides information about the number of Web connections, throughput volume, HTTP responses, server retries, and downloaded pages during the session step. For more information, see Chapter 28, "Web Resource Monitoring."

The **System Resource** monitors gauge the Windows, UNIX, Tuxedo, SNMP, and Antara FlameThrower resources used during a session step. For more information, see Chapter 29, "System Resource Monitoring."

The **Network Delay** monitor displays information about the network delays on your system. For more information, see Chapter 30, "Network Monitoring."

The **Firewall** monitor measures statistics at the firewall servers during the session step. For more information, see Chapter 31, "Firewall Server Performance Monitoring."

The **Web Server Resource** monitors measure statistics at the Apache, Microsoft IIS, iPlanet (SNMP) and iPlanet/Netscape Web servers during the session step. For more information, see Chapter 32, "Web Server Resource Monitoring."

The **Web Application Server Resource** monitors measure statistics at the Web application server(s) during the session step. For more information, see Chapter 33, "Web Application Server Resource Monitoring."

The **Database Server Resource** monitors measure statistics related to the SQL server, Oracle, Sybase, and DB2 databases. For more information, see Chapter 34, "Database Resource Monitoring."

The **Streaming Media** monitors measure statistics at the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer client. For more information, see Chapter 35, "Streaming Media Monitoring."

The **ERP/CRM Server Resource** monitor measures statistics at the ERP servers during the session step. For more information, see Chapter 36, "ERP/CRM Server Resource Monitoring."

The **Java Performance** monitors measure statistics of Java 2 Platform, Enterprise Edition (J2EE) objects, Enterprise Java Bean (EJB) objects and Java-based applications, using J2EE and EJB machines. For more information, see Chapter 37, "Java Performance Monitoring", and Chapter 38, "J2EE Performance Monitoring."

The **Application Deployment Solutions** monitor measures statistics of the Citrix MetaFrame XP and 1.8 servers during a session step run. For more information, see Chapter 39, "Application Deployment Solution Monitoring."

The **Middleware Performance** monitors measure statistics of the Tuxedo and IBM WebSphere MQ servers during a session step run. For more information, see Chapter 40, "Middleware Performance Monitoring."

The **Application Traffic Management** monitor measures statistics of the F5 BIG-IP server during a session step run. For more information, see Chapter 41, "Application Traffic Management Monitoring."

The **Security** monitor displays information about simulated attacks on the server during a session step using the Distributed Denial of Service graph. For more information, see Chapter 42, "Security Monitoring."

The **SiteScope Server** monitor measures the SiteScope resources used during a session step. For more information, see Chapter 43, "SiteScope Server Monitoring."

You view the data collected by the monitors using the LoadRunner Tuning Module online graphs. For information on configuring graph settings and measurements, and exporting graph data, see Chapter 26, "Configuring Online Graphs."

All of the monitors allow you to view a summary of the collected data at the conclusion of the session step. Using LoadRunner Analysis, you can generate a graph for any of the monitors. For more information, refer to the *LoadRunner Analysis User's Guide*.

Note: For a detailed list of LoadRunner Tuning Module's monitors, see Mercury Interactive's Web site (http://www.mercuryinteractive.com/products/loadrunner/load_testing_monitors/supported.html).

Choosing Monitors and Measurements

You can select the measurements to monitor for each of your servers via the Monitors button on the main toolbar or the Element Monitors tab in the System Topology window.

Note that you select the measurements to monitor the topology elements—not to monitor the physical hosts. If you map the same physical host to more than one topology element, you will typically be interested in monitoring those measurements that are relevant to each element. For example, if the physical host is mapped to a Web server element and a database server element, you can monitor the Web-related measurements on the Web server, and the database-related ones on the database server.

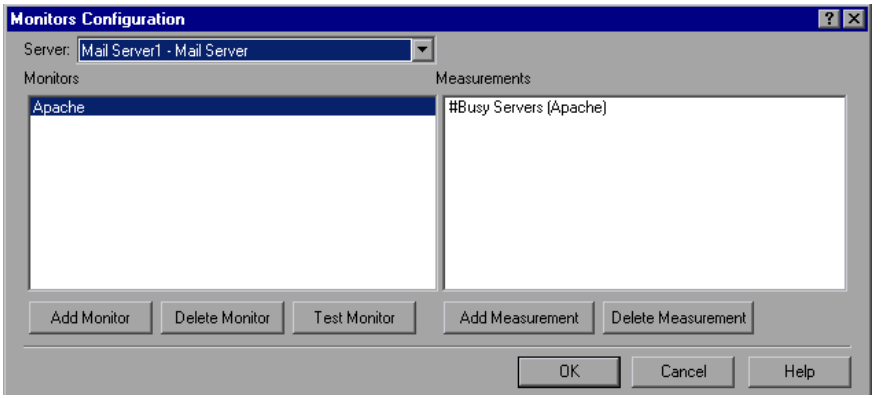
The relevant measurements will appear on the graphs belonging to the individual topology elements.

To select measurements to monitor via the System Topology window:

➤ Follow the directions in "Selecting Monitors Manually," on page 40.

To select measurements to monitor via the Monitors button on the main toolbar:

1 Click **Monitors**. The Monitors Configuration dialog box is displayed:



2 Select the server whose monitors you want to configure from the list box.

3 To add measurements to monitor, click **Add Monitor**. The **Select Measurements to Monitor** dialog box is displayed. Choose the monitor and the measurements for the specific server.

Note: For detailed instructions on using the **Select Measurements to Monitor** dialog box, see "Selecting Monitors Manually," on page 40.

---

4 Click **OK** to save your configuration.

Understanding the Monitors Configuration Dialog Box

The Monitors Configuration dialog box lets you select the measurements to monitor for each of your servers.

Server: Select the server whose monitors you want to configure from the list box.

Monitors: The monitors that are currently assigned to monitor the specified server.

Measurements: When you click on a monitor in the Monitors pane, the measurements that have been specified to be monitored by that monitor are listed in the Measurements pane.

Add Monitor: Opens the Select Measurements to Monitor dialog box.

Delete Monitor: Deletes the selected monitor from the Monitors pane.

Add Measurement: Adds a measurement to the list in the Measurements pane.

Delete Measurement: Deletes the selected measurement from the list of measurements in the Measurements pane.

Test Monitor: This button appears for some monitors. Tests whether you can access the monitor.

Viewing the Monitors

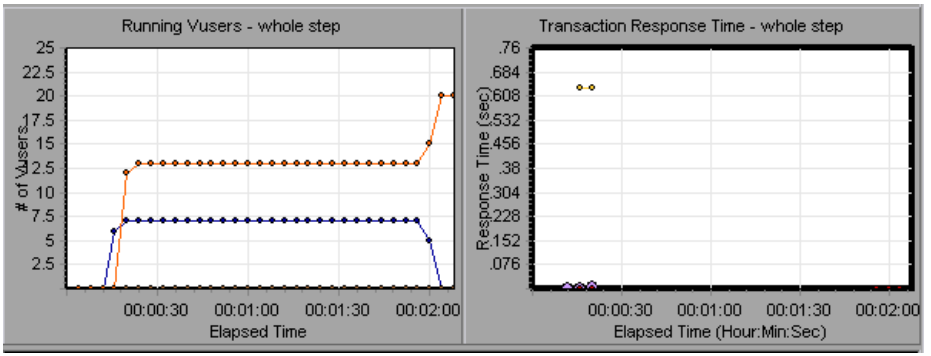
The online monitors start operating as soon as you assign them to an element, even before you start a session step. The measurements from each monitor are displayed on a separate graph.

Note: The Runtime, Transaction and Web Resource graph groups are session step dependent: they start displaying measurements only after you start a session step.

To view the online monitors and session step dependent graphs:

1 Start the session step by clicking the **Execute** button. Alternatively, you can choose **Session** > **Start Time**, select one of the start step options, and click **OK**.

2 Click the **Session** tab. The default graphs are displayed below the Session Step Groups window.



3 Double-click a graph to maximize it. Repeat the operation to restore the tiled view.

Note: The Transaction Monitor graphs will not contain any data unless transactions are being executed. In addition, the other graphs will not contain any data unless you set up a list of resources to monitor before running your session step.

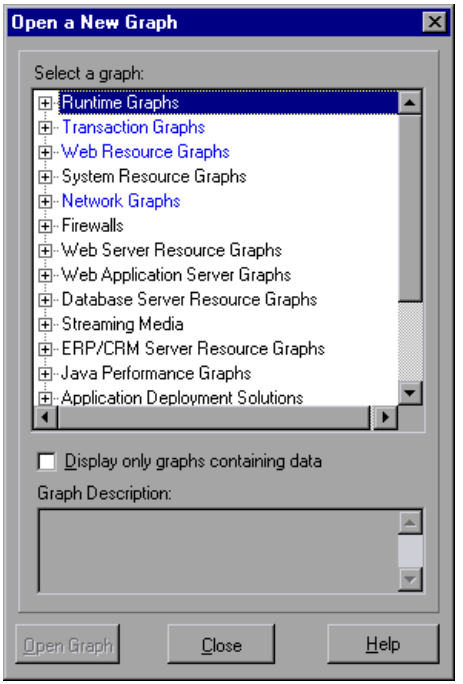
Opening Online Monitor Graphs

By default, LoadRunner Tuning Module displays two graphs in the Execute view: Running Vusers and Transaction Response Time. You can display the other graphs by using one of the following methods:

- Opening a new graph using the Open a New Graph dialog box
- Using the Select Online Graphs dialog box

To open a new graph using the Open a New Graph dialog box:

1 Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.

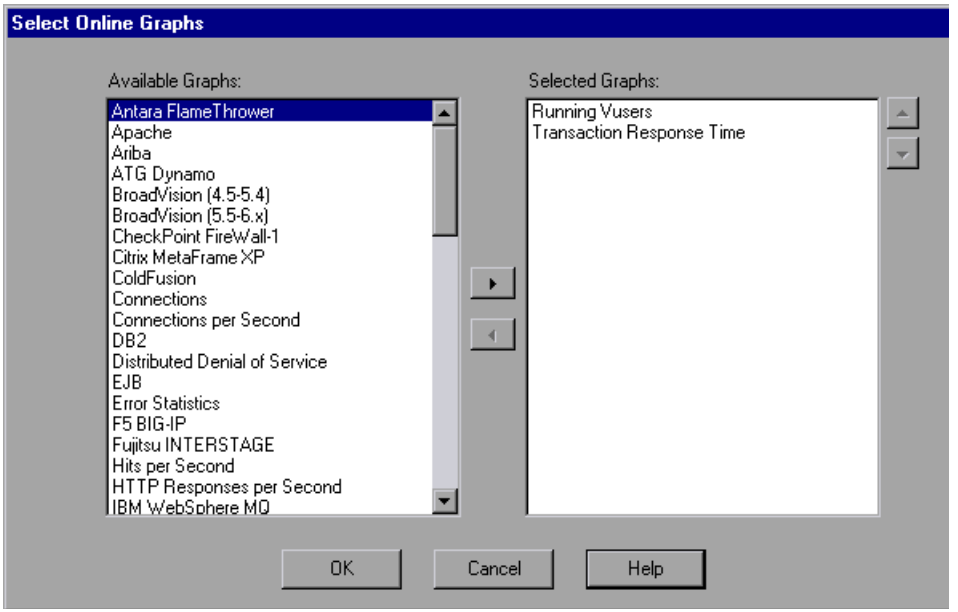


2 Click the "+" in the left pane to expand the graph tree, and select a graph. You can view a description of the graph in the **Graph Description** box. To view only graphs which contain data, select the **Display only graphs containing data** check box.

3 Click **Open Graph**, or Drag the selected graph into the right pane of the Session view. The graph appears in the graph view area.

To specify the graphs via the Select Online Graphs dialog box:

1 Click the **Graphs** button in the toolbar. The Select Online Graphs dialog box opens.



The Selected Graphs pane shows the graphs that are currently selected to be displayed in the Execute view. If you have not specified any graphs, the pane lists the two default graphs.

2 Add the required graphs to the Selected Graphs pane, and use the up- and down-arrows to determine the graphs' positions in the Session tab.

Understanding the Open a New Graph Dialog Box

The Open a New Graph dialog box enables you to open a new graph and view its description.

Select a graph: Click the "+" to the left of each category to expand the tree view. Select a graph.

Note: You can only open one graph at a time.

Display only graphs containing data: Select this option to view graphs which contain data only. To view the entire list of LoadRunner Tuning Module Analysis graphs, clear this option.

Graph description: Displays the selected graph's description.

Open Graph: Opens the selected graph and displays it in the Graph Tree View.

Understanding the Select Online Graphs Dialog Box

The Select Online Graphs dialog box allows you to specify the graphs that will be displayed in the **Session** tab, and the position that each graph will occupy.

The graphs are specific to the step for which you define them. When you save a session and subsequently reopen it, these are the graphs that are displayed for the step.

To select a graph for display:

➤ Click the graph in the Available Graphs section and then click the right-arrow to move the graph to the Selected Graphs section.

To remove a graph from the list of graphs to display:

➤ Click the graph in the Selected Graphs section and then click the left-arrow. The graph is removed from the Selected Graphs section.

To change the order in which the graphs are displayed in the Session tab:

➤ The graph at the top of the list in the Selected Graphs pane will be displayed in the top row in the left-most position. The graph at the bottom of the list will be displayed in the bottom row in the right-most position. Select a graph in the Selected Graphs section and then use the up- and down-arrows to change the graph's position. Repeat this process with the other graphs until you have positioned the graphs correctly.

Customizing the Online Monitor Display View

LoadRunner Tuning Module lets you display up to 16 online monitor graphs simultaneously.

To customize your online monitor display:

Right-click a graph, select **View Graphs**, and then select the number of graphs you want to view. You can choose from **Show One Graph**, **Show Two Graphs**, **Show Four Graphs**, **Show Eight Graphs**, or **Custom Number**. If you select **Custom Number**, enter the number of graphs you want to view in the View Graphs dialog box, and click **OK**. The number of graphs selected open in the graph view area.

To display only one graph, double-click the graph pane. To return to the previous view, double-click the graph again.

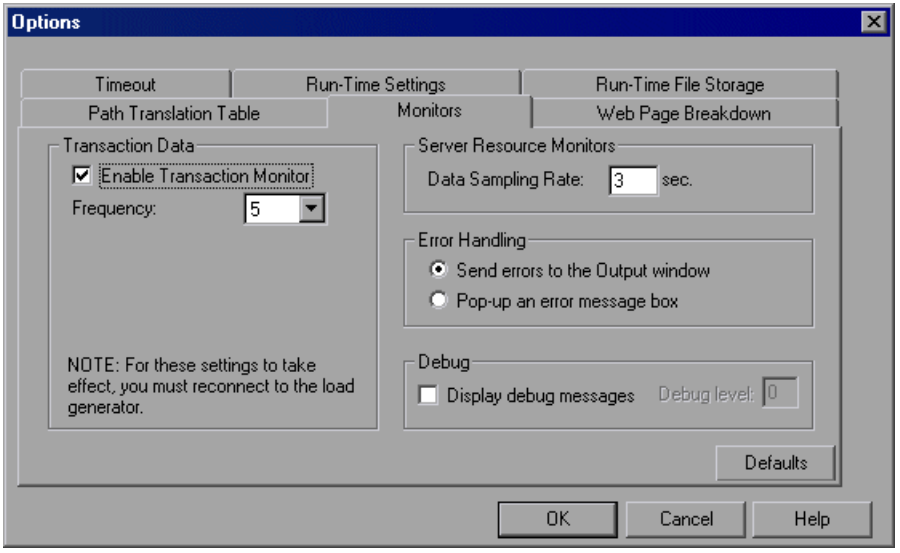
Setting Monitor Options

Before running your session step, LoadRunner Tuning Module lets you configure the settings for your online monitors. You can set the data sampling rate, error handling, debugging, and frequency settings for the online monitors.

When you save a session step, the online monitor configuration settings are saved as well.

To set monitor options:

1 Select **Tools** > **Options** and select the **Monitors** tab.



2 Select **Enable Transaction Monitor**, and specify the frequency at which the monitor should send updates to the Console for the Transaction, Data Point, and Web Resource graphs.

Note: You cannot modify these settings during session step execution; you must stop the session step before disabling the monitor or changing its frequency.

---

3 Enter a sampling rate.

4 Set the desired **Error Handling** option.

5 To display debug messages in the Output window, select the **Display Debug Messages** check box. For the Network monitor, specify a **Debug level** from 1-9.

6 Click **OK** to save your settings and close the Options dialog box.

You can configure an additional monitor setting while working in Expert mode. For information on working in Expert mode, see Appendix C, "Working in Expert Mode."

Understanding the Options - Monitors Tab

The Monitors Tab lets you enable the Transaction monitor, configure the behavior of the transaction data, and set the data sampling rate, error handling, debugging, and frequency settings for the online monitors.

Transaction Data: Configures the behavior of data for the Transaction, Data Point, and Web Resource online graphs.

- **Enable Transaction Monitor:** Enables the online Vuser Transaction monitor to begin monitoring transactions at the start of a session step.
- **Frequency:** Select the frequency, in seconds, at which the online monitor samples the data to produce the Transaction, Data Point, and Web Resource online graphs. The default is 5 seconds. For a small session step, it is recommended that you use a frequency of 1. For a large session step, it is recommended that you use a frequency of 3-5. The higher the frequency, the less network traffic there will be. The data is averaged for the frequency period defined, and only one value is sent to the Console.

  For information on enabling and disabling the Transaction monitor and Web page breakdown, see Chapter 27, "Run-Time and Transaction Monitoring."

Server Resource Monitors: Configures the behavior of the Server Resource monitors.

➤ **Data Sampling Rate:** The sampling rate is the period of time (in seconds) between consecutive samples. Enter the rate at which LoadRunner Tuning Module samples the session step for monitoring data. By default, the online monitor samples the data at intervals of three seconds. If you increase the sampling rate, the data is monitored less frequently. This setting applies to all graphs. To set a sampling rate for a specific graph, see Chapter 26, "Configuring Graph Properties.".

Note: The data sampling rate you set is applied to all server monitors that you subsequently activate. It is not applied to server monitors that have already been activated. To apply the new data sampling rate to activated server monitors, save your session step and reopen it.

Each monitor has a different minimum sampling rate. If the default sampling rate, or the rate set in the Options Monitors tab is less than a monitor's minimum sampling rate, the monitor will sample data at intervals of its minimum sampling rate. For example, the minimum sampling rate for the Oracle Monitor is 10 seconds. If the sampling rate in the Options Monitors tab is set at less than 10 seconds, the Oracle Monitor will continue to monitor data at 10 second intervals.

Error Handling: Controls the way in which LoadRunner Tuning Module issues error messages. Select one of the following options:

➤ **Send errors to the Output window:** Sends all errors to the Output window (default).

➤ **Pop-up an error message box:** Sends errors to a message box . To dismiss the message box, the operator must click **OK**.

Debug: For debugging a session step, you can set the following option:

➤ **Display debug messages:** Sends debug-related messages to the output log. You can also specify a debug level from 1-9. The debug level is only relevant to the Network monitor.

26

Configuring Online Graphs

You can view the data collected by the monitors using the LoadRunner Tuning Module online monitor graphs.

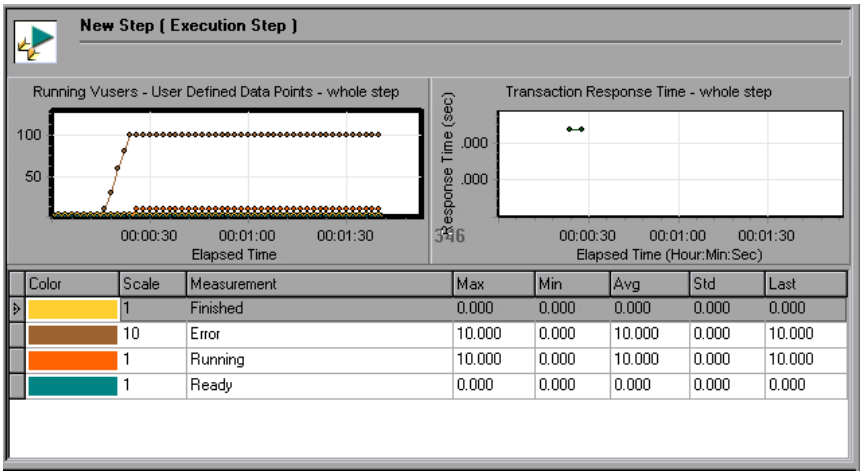
- About Online Monitor Graphs
- Configuring Graph Properties
- Configuring Graph Measurements
- Merging Graphs
- Adding Measurement Alerts
- Exporting Online Monitor Graphs
- Viewing Data Offline

About Online Monitor Graphs

Data generated during session step execution is displayed in the LoadRunner Tuning Module online monitor graphs. Information about the measurements is listed in the legend below the graph. Each value is represented by a colored line. The legend displays the measurements for the selected graph only.

For more information on opening monitor graphs and customizing the display, see "Opening Online Monitor Graphs" on page 295, and

"Customizing the Online Monitor Display View" on page 299.



To get additional information about a measurement, right-click the measurement and choose **Description**.

To focus on a particular line, you can:

- **Highlight a measurement:** To highlight a specific measurement, select it in the legend. The corresponding line in the graph is displayed in bold.
- **Hide a measurement:** To hide a measurement, right-click the measurement and choose **Hide**. To hide all measurements other than the measurement selected, right-click the measurement and choose **Show Only Selected**. To show a hidden measurement, right-click the measurement and choose **Show**.
- **Pause the monitor:** To pause a specific graph during session step execution, right-click the graph and select **Freeze**. To resume, repeat the above action. When you resume, the graph displays the data for the paused period.

To maintain the sort order after the legend is refreshed, right-click the graph and select **Keep Legend Sorted**. Click again to remove the sort order. The legend is refreshed every five seconds.

Configuring Graph Properties

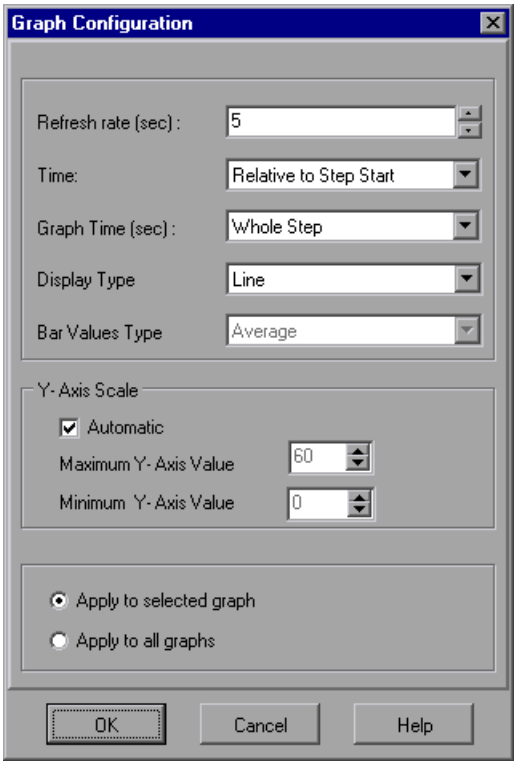
LoadRunner Tuning Module lets you configure the settings for your online monitor graphs. You can customize your graph in the following areas:

- Refresh Rate
- Time
- Graph Time
- Display Type
- Bar Value
- Y-Axis Style
- Network Delay View

Note that these settings can be set globally—to apply to all graphs—or per graph.

To customize your graphs:

1 Right-click the online graph you want to configure (in either the right or left pane) and select **Configure**. The Graph Configuration dialog box opens.



2. To apply the dialog box settings to all graphs, select **Apply to all graphs**.
3. Enter the desired refresh rate—the time between graph updates—in the Refresh Rate box.
4. Select a style for the x-axis from the Time box.
5. Select a value from the Graph Time box. The graph time is the time in seconds displayed by the x-axis.
6. Select a graph style from the Display Type box.
7. If the selected display type is Bar, choose a value from the Bar Values Type box. This determines the type of value that will be displayed in the bar graph. You can choose between **Average**, **Last Value**, **Minimum** and **Maximum**.
8. Select a maximum or minimum value for the y-axis, or choose **Automatic** to view graphs using the default y-axis scale.
9. Choose to apply configuration settings to all graphs or selected graphs only (not available for the Network Delay Time graph).
10. For the Network Delay Time graph, you can select the following options:
    - **SubPaths**: Displays the delay measurements from the source machine to each of the nodes along the network path.
    - **DNS name:** Displays the DNS names of the measurements in the legend.
11. Click **OK** to save your settings and close the Graph Configuration dialog box.

Understanding the Graph Configuration Dialog Box

The Graph Configuration dialog box lets you customize the online graph settings.

Refresh Rate

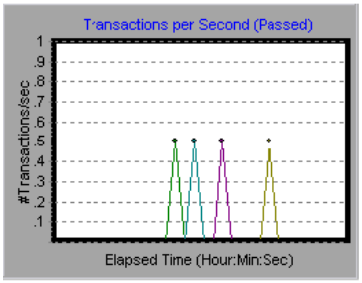
The interval in which the graph is refreshed with new data. By default, the graph is refreshed every five seconds. If you increase the refresh rate, the data is refreshed less frequently. Note that in a large load test, it is recommended to use a refresh rate of three to five seconds. This enables you to avoid problems with CPU resource usage.

Time

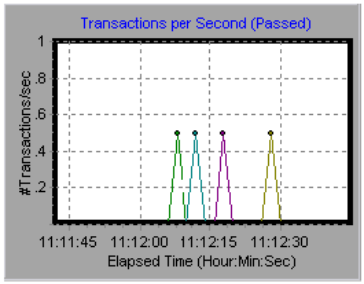
You can specify how the graph displays the x-axis time.

- **Don't Show:** Instructs LoadRunner Tuning Module not to display values for the x-axis.
- **Clock Time**: Displays the absolute time, based on the system clock.
- **Relative to Session Start**: Displays the time relative to the beginning of the session step. Note that if no step is running, clock time is displayed.

In the following example, the graph is shown with the **Don't Show** and **Clock Time** options. In the following example, the graph is shown with the *Don't Show* and *Clock Time* options:



Don't Show§



Clock Time§

Graph Time

Indicate the scale for a graph's x-axis when it is time-based. A graph can show 60 or 3600 seconds of activity. To see the graph in greater detail, decrease the graph time. To view the performance over a longer period of time, increase the graph time. The available graph times are: **Whole Step**, **60**, **180**, **600**, and **3600** seconds.

Display Type

Specify whether LoadRunner Tuning Module displays a graph as a line graph or a bar graph. By default, each graph is displayed as a line graph. Note that for the Network Delay graph, if you select **View Segments**, you can view the network segments of the graph as an area graph or a pie graph.

Bar Value

Choose a value from the Bar Values Type box (if the selected display type is Bar). This determines the type of value that will be displayed in the bar graph. You can choose between **Average**, **Last Value**, **Minimum** and **Maximum**.

Y-Axis Style

Instruct LoadRunner Tuning Module to display graphs using the default y-axis scale, or you specify a different y-axis scale. Click **Automatic** if you want LoadRunner Tuning Module to use the default y-axis values. Specify a maximum or minimum value for the y-axis if you want to modify the y-axis scale.

Network Delay View

This option only appears when you configure the Network Delay Time graph. Click **SubPaths** to view the delay measurements from the source machine to each of the nodes along the network path. Click **DNS name** to view the DNS names of the measurements displayed in the legend.

Configuring Graph Measurements

You can configure the following online measurement settings:

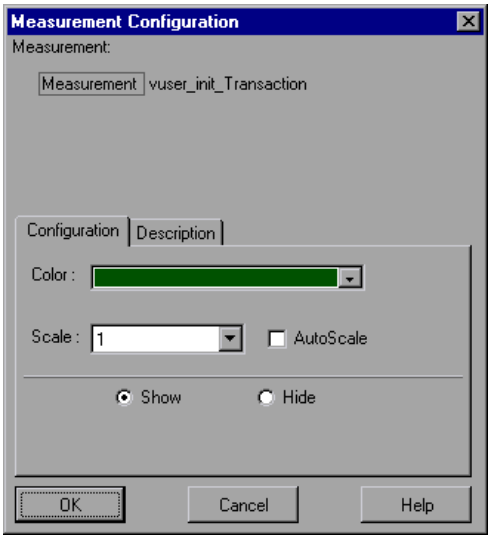
- Change Line Colors
- Set Measurement Scale
- Show and Hide Transactions

Changing Line Colors

LoadRunner Tuning Module assigns a unique color to each measurement. You can modify the color using the configuration interface.

To change the line color of a measurement:

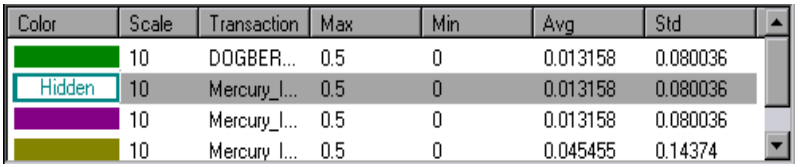
1 In the legend below the graphs, select the measurement you want to configure. Right-click and choose **Configure**. The Measurement Configuration dialog box opens.



2 To change the color of the line, select a color from the Color list.

3 Click **OK** to accept the settings and close the dialog box.

The specified color changes are reflected in the graph and in the legend beneath the graph. The color is displayed in the first column of the legend.



Setting the Scale of the Measurement

You can modify the scale of a measurement—the relationship between the y-axis and the graph's actual value. For example, a scale set at 1 indicates that the measurement's value is the value of the y-axis. If you choose a scale

of 10, you must divide the y-axis value by 10 to obtain the true value of the measurement.

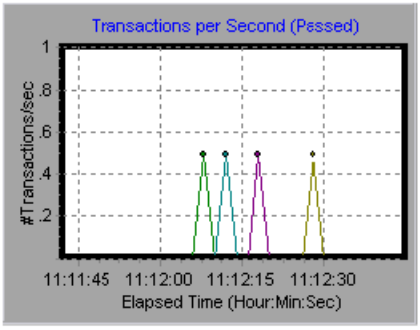
To set the scale of a measurement:

1 Select the measurement you want to configure. Right-click and choose **Configure**. The Measurement Configuration dialog box opens.

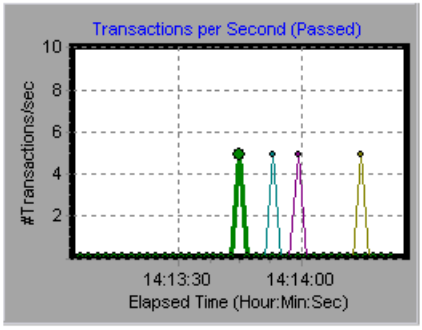
2 Clear the **Autoscale** check box and select the desired ratio from the Scale list.

3 Click **OK** to accept the settings and close the dialog box.

In the following example, the same graph is displayed with a scale of 1 and 10.



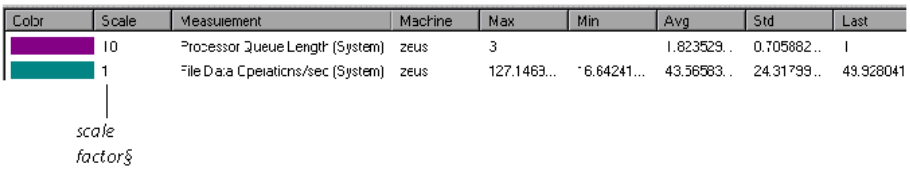
scale = 1



scale = 10

The actual graph values range from 0-1, as shown in the left graph. You can view the information more accurately using a larger scale for the display, as shown in the right graph. However, to obtain the actual values, you need to divide the displayed value by the scale. In the example above, the highest value shown in the graph is 5. Since the scale is 10, the actual value is 0.5.

The legend below the graph indicates the scale factor.



By default, LoadRunner Tuning Module uses the *autoscale* option, which automatically scales the measurements by calculating the best ratio for displaying the graph.

Showing and Hiding Transactions

By default, the Transaction Monitor displays a line for each item in the transaction list. You can hide the line for any of the monitored transactions in order to focus on a specific measurement.

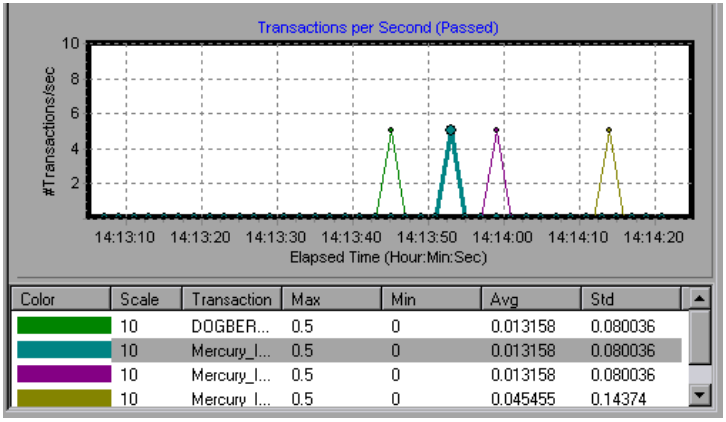
To show or hide a transaction:

1 To hide a measurement, click **Hide**. To show a hidden resource, click **Show**. To hide all measurements other than the measurement selected, right-click the measurement and choose **Show Only Selected**.

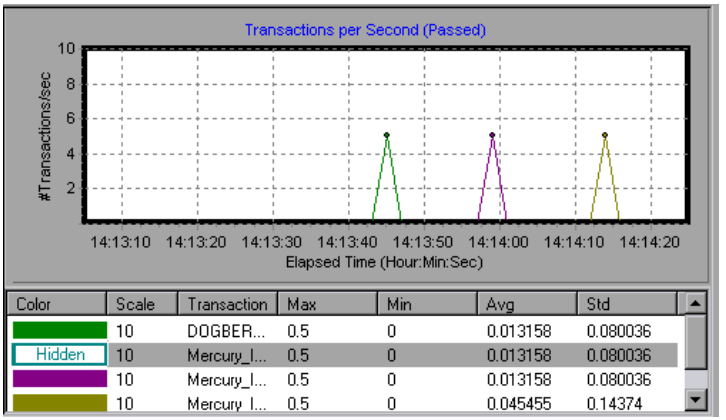
2 Click **OK** to accept the settings and close the dialog box.

Note that you can also show and hide measurements without opening the Measurement Configuration dialog box, by right-clicking a measurement in the legend and selecting **Show/Hide**.

In the following example, a line is shown for each measurement



In this example, the second item in the legend is hidden.



Understanding the Measurement Configuration - Configuration Tab

The Measurement Configuration tab lets you change line colors, set the scale of a measurement, and show or hide transactions.

Measurement: Displays the type of resource being monitored.

Machine: Displays the name of the machine whose resources are being monitored (appears only in cases where a machine's resources are being monitored).

Note: When monitoring a network path, the **Network Type** will appear here instead of Machine.

Color: Select a color to be assigned to the selected measurement.

Scale: Displays the relationship between the y-axis and the graph's actual value. For example, a scale set at 1 indicates that the measurement's value is the value of the y-axis. If you choose a scale of 10, you must multiply the y-axis value by 10 to obtain the true value of the measurement.

Autoscale: Instructs LoadRunner Tuning Module to automatically scale the measurement by calculating the best ratio for displaying the graph. For some graphs, this option is not available.

Show: Shows the selected resource. The line for the selected resource reappears in the graph. By default, all resource measurements are displayed in the chart.

Hide: Hides the selected resource. The line for the selected resource disappears from the graph. The hidden resources are displayed as unfilled boxes.

Understanding the Measurement Configuration - Description Tab

The Measurement Description tab displays information about the measurement.

Measurement: Displays the type of resource being monitored.

Machine: Displays the name of the machine whose resources are being monitored (appears only in cases where a machine's resources are being monitored).

Description: Displays a description of the selected measurement.

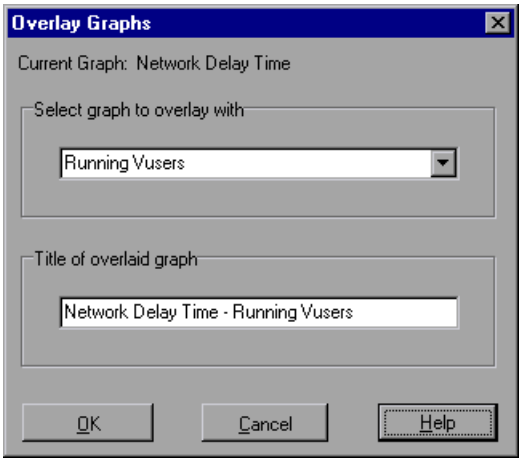
Merging Graphs

LoadRunner Tuning Module lets you merge the results of two graphs from the same session step into a single graph. The merging allows you to compare several different measurements at once. For example, you can make a merged graph to display the Web Throughput and Hits per Second, as a function of the elapsed time. Note that in order to merge graphs, their x-axis must be the same measurement.

When you overlay the contents of two graphs that share a common x-axis, the left y-axis on the merged graph shows the current graph's values. The right y-axis shows the values of the graph that was merged.

To overlay two graphs:

1 Right-click one of the graphs you want to overlay, and select **Overlay Graphs**. The Overlay Graphs dialog box opens.



2 Select a graph with which you want to overlay the current graph. The drop-down list only shows the active graphs that have a common x-axis with the current graph.

3 Enter a title for the overlaid graph.

4 Click **OK**. The merged graph appears in the graph view area.

Adding Measurement Alerts

LoadRunner Tuning Module allows you to add and remove alert conditions for graph measurements.

To add/remove an alert to a graph measurement:

1 Right-click the measurement and choose one of the following:

- **Alerts:** Opens the Alerts dialog box.
- **Alert on <Measurement Name>:** Opens the Alerts dialog box and displays the name of the selected measurement in the Available Alerts list.

2 Configure the alert properties and click **OK**.

3 If an alert is triggered, the measurement blinks in the graph and in the legend. To stop the measurement blinking, right-click the measurement and choose **Stop Highlight**.

4 To disable an alert, clear the check box of the relevant alert in the Available Alerts list.

For more information on configuring alerts, see Chapter 13, "Defining Alerts."

Exporting Online Monitor Graphs

LoadRunner Tuning Module allows you to export an online graph to HTML for viewing at a later stage. When you export to HTML, the legend is also displayed with the graph.

To export an online graph to HTML:

1 Right-click the graph you want to export and choose **Export to HTML**. The Select Filename and Path dialog box opens.

2 Specify a filename and path and click **Save**.

Viewing Data Offline

After monitoring resources during a session step run, you can view a graph of the data that was gathered using the LoadRunner Tuning Module Analysis. When you run the Analysis utility, it processes the data and generates a graph for each measurement that was monitored.

To view a graph, choose **Graph > Add Graph** in the Analysis window. For more information about working with the LoadRunner Tuning Module Analysis at the conclusion of the session step, refer to the *LoadRunner Analysis User's Guide*.

27

Run-Time and Transaction Monitoring

While running a session step, you can use LoadRunner Tuning Module's Run-Time and Transaction monitors to view graphs of run-time status and transaction performance.

This chapter describes:

- Run-Time Graphs
- User-Defined Data Points Graph
- Transaction Monitor Graphs
- Enabling the Transaction Monitor
- Adding Transactions to a Script
- Enabling Web Page Breakdown

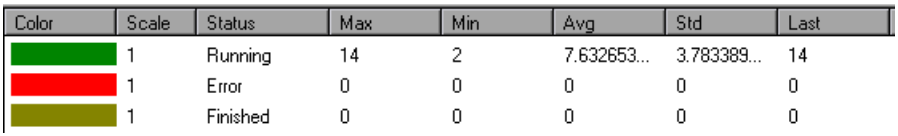
About Run-Time and Transaction Graphs

The *Run-Time* monitor provides information about the status of the Vusers participating in the session step, and the number and types of errors that the Vusers generate. In addition, the Run-Time monitor provides the User-Defined Data Points graph, which displays the real time values for user-defined points in a Vuser script.

The *Transaction* monitor displays the transaction rate and response time during session step execution. For more information about transactions, see "Adding Transactions to a Script" on page 326.

Run-Time Graphs

The monitor's **Running Vusers** graph provides information about the status of the Vusers running in the current session step on all load generator machines. The graph shows the number of running Vusers, while the information in the legend indicates the number of Vusers in each state.



Color	Scale	Status	Max	Min	Avg	Std	Last
	1	Running	14	2	7.632653...	3.783389...	14
	1	Error	0	0	0	0	0
	1	Finished	0	0	0	0	0

The Status field of each Vuser displays the current status of the Vuser. The following table describes each Vuser status.

Status	Description
RUNNING	The total number of Vusers currently running on all load generators.
READY	The number of Vusers that completed the initialization section of the script and are ready to run.
FINISHED	The number of Vusers that have finished running. This includes both Vusers that passed and failed.
ERROR	The number of Vusers whose execution generated an error. Check the Status field in the Vuser view or the Output window for a complete explanation of the error.

The monitor's **Error Statistics** graph provides details about the number of errors that accrue during each second of the session step run. The errors are grouped by error source—for example, the location in the script or the load generator name.

The **Vusers with Error Statistics** graph provides details about the number of Vusers that generate errors during session step execution. The errors are grouped by error source.

User-Defined Data Points Graph

The **User-Defined Data Points** graph displays the real-time values of user-defined data points. You define a data point in your Vuser script by inserting an **lr_user_data_point** function at the appropriate place (**user_data_point** for GUI Vusers and **lr.user_data_point** for Java Vusers).

```
Action1()
{
	lr_think_time(1);
	lr_user_data_point ("data_point_1",1);
	lr_user_data_point ("data_point_2",2);
	return 0;
}
```

For Vuser protocols that support the graphical script representations such as Web and Oracle NCA, you insert a data point as a User Defined step. Data point information is gathered each time the script executes the function or step. For more information about data points, refer to the *Online Function Reference*.

By default, LoadRunner Tuning Module displays all of the data points in a single graph. The legend provides information about each data point. If desired, you can hide specific data points using the legend below the graphs.

You can also view data points offline, after the completion of the session step. For more information, refer to the *LoadRunner Analysis User's Guide*.

Transaction Monitor Graphs

The *Transaction* monitor provides the following graphs:

- Transaction Response Time
- Transactions per Second (Passed)
- Transactions per Second (Failed, Stopped)
- Total Transactions per Second (Passed)
- Transaction Count (Passed)
- Transaction Count (Failed, Stopped)

The **Transaction Response Time** graph shows the average response time of transactions in seconds (y-axis) as a function of the elapsed time in the session step (x-axis).

The **Transactions per Second (Passed)** graph shows the number of successful transactions performed per second (y-axis) as a function of the elapsed time in the session step (x-axis).

The **Transactions per Second (Failed, Stopped)** graph shows the number of failed and stopped transactions per second (y-axis) as a function of the elapsed time in the session step (x-axis).

The **Total Transactions per Second (Passed)** graph shows the total number of completed, successful transactions per second (y-axis) as a function of the elapsed time in the session step (x-axis).

The **Transaction Count (Passed)** graph shows the total number of completed, successful transactions since the beginning of the session step.

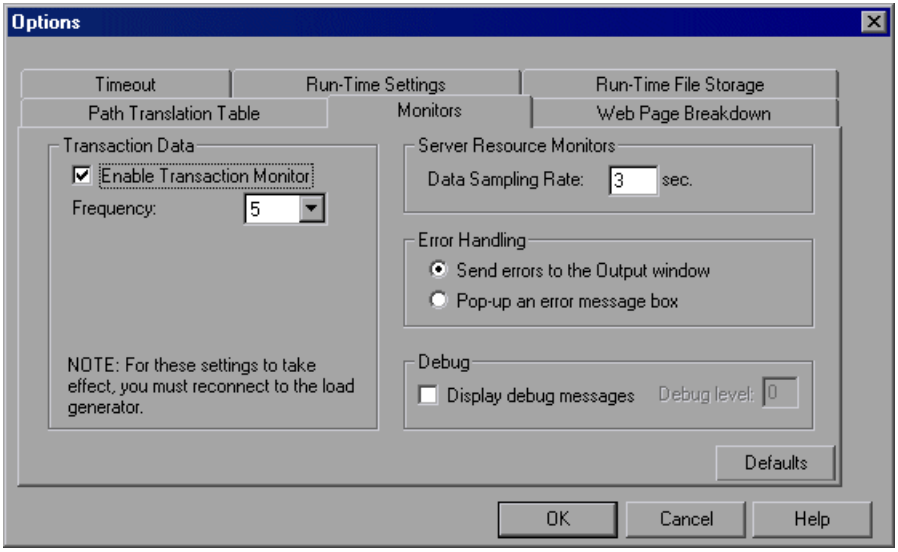
The **Transaction Count (Failed, Stopped)** graph shows the total number of transactions that failed or were stopped since the beginning of the session step.

Enabling the Transaction Monitor

The Transaction monitor is enabled by default—it automatically begins monitoring Vuser transactions at the start of a session step. You can disable the Transaction monitor in order to conserve resources.

To enable the Transaction monitor:

1 Choose **Tools** > **Options** and select the **Monitors** tab.



2 Enable transaction monitoring by selecting the **Enable Transaction Monitor** check box. To disable transaction monitoring, clear the **Enable Transaction Monitor** check box.

Adding Transactions to a Script

If there are no transactions defined in your Vuser script, no data will be displayed in the online graphs. To add transactions to an existing script, edit it using the appropriate tool. The following table shows the script generation tools for each script type:

Script type	Editing tool
GUI Windows	WinRunner
non-GUI Windows	VuGen (Vuser Generator)
SAP	QuickTest for SAP

To add a transaction to a script:

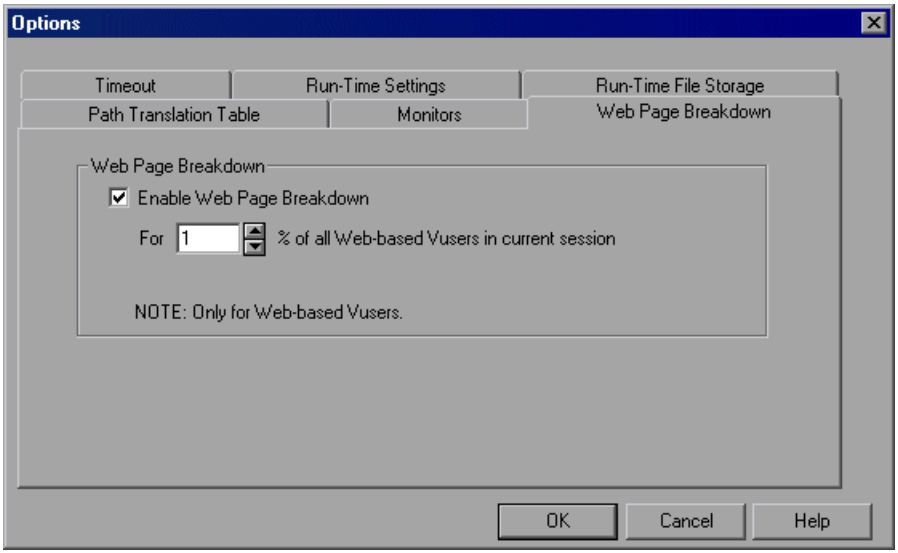
1. Right-click the script you want to edit in the session tree, and select **View Script**. The script generation tool opens.
2. Insert Start and End Transaction functions or markers throughout your script.

   For more information, refer to the appropriate user's guide as described in the *Welcome* chapter.

Enabling Web Page Breakdown

In order for the Analysis to generate Web Page Breakdown graphs, which provide you with performance information for each transaction and sub-transaction defined in your script, you must enable the Web page breakdown feature in the Console before running your session step.

To enable Web page breakdown, choose **Tools** > **Options** and select the **Web Page Breakdown** tab.



Web Page Breakdown

➤ **Enable Web Page Breakdown:** Enables you to view the Web Page Breakdown graphs in the Analysis.

➤ **For X % of all Vusers in current session:** Specify the percentage of Vusers for which you want transaction breakdown to be performed.

For more information about Web Page Breakdown graphs, refer to the *LoadRunner Analysis User's Guide*.

28

Web Resource Monitoring

You can obtain information about the performance of your Web server using LoadRunner Tuning Module's Web Resource monitor.

This chapter describes:

- Hits per Second Graph
- Throughput Graph
- HTTP Responses per Second Graph
- Pages Downloaded per Second Graph
- Retries per Second Graph
- Connections Graph
- Connections per Second Graph
- SSL Connections per Second Graph

About Web Resource Monitoring

The Web Resource monitor enables you to analyze the throughput on the Web server, the number of hits per second that occurred during the session, the number of HTTP responses per second, the HTTP status codes (which indicate the status of HTTP requests, for example, "the request was successful," "the page was not found") returned from the Web server, the number of downloaded pages per second, and the number of server retries per second.

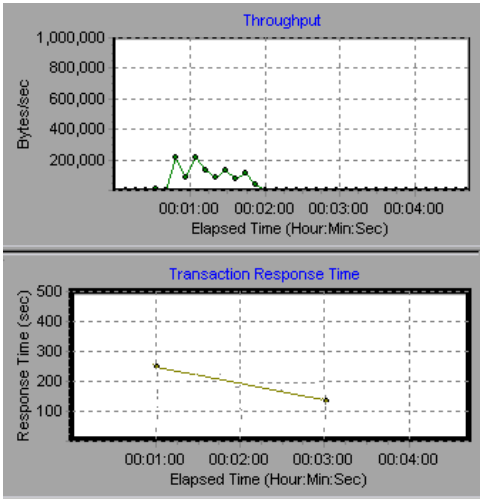
Hits per Second Graph

The **Hits Per Second** graph shows the number of hits (HTTP requests) to the Web server (y-axis) as a function of the elapsed time in the session step (x-axis). This graph can display the whole step, or the last 60, 180, 600, or 3600 seconds. You can compare this graph to the Transaction Response Time graph to see how the number of hits affects transaction performance.

Throughput Graph

The **Throughput** graph shows the amount of throughput on the Web server (y-axis) during each second of the session step run (x-axis). Throughput is measured in bytes and represents the amount of data that the Vusers received from the server at any given second. You can compare this graph to the Transaction Response Time graph to see how the throughput affects transaction performance.

In the following example, the Transaction Response time graph is compared with the Throughput graph. It is apparent from the graph that as the throughput decreases, the transaction response time also decreases. The peak throughput occurred at approximately 1 minute into the step. The highest response time also occurred at this time.



HTTP Responses per Second Graph

The **HTTP Responses per Second** graph shows the number of HTTP status codes—which indicate the status of HTTP requests, for example, “the request was successful,” “the page was not found”—(y-axis) returned from the Web server during each second of the session step run (x-axis), grouped by status code. You can group the results shown in this graph by script (using the "Group By" function) to locate scripts which generated error codes.

The following table displays a list of HTTP status codes:

Code	Description
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized

Code	Description
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request - URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version not supported

For more information on the above status codes and their descriptions, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.

Pages Downloaded per Second Graph

The **Pages Downloaded per Second** graph shows the number of Web pages (y-axis) downloaded from the server during each second of the session step run (x-axis). This graph helps you evaluate the amount of load Vusers generate, in terms of the number of pages downloaded.

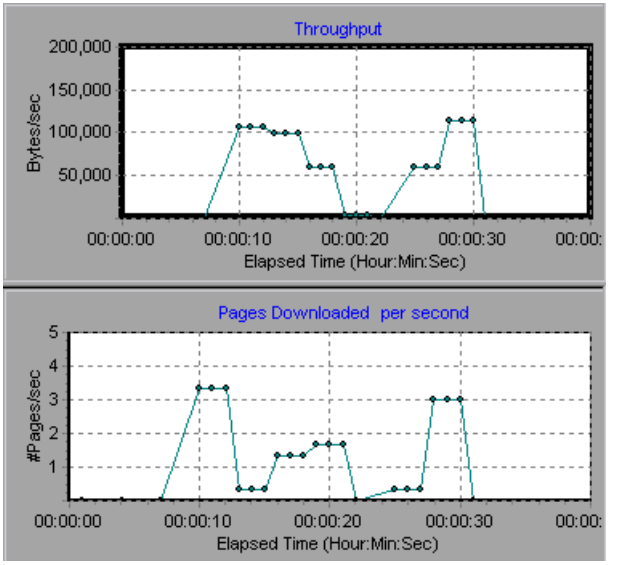
Note: In order to view the Pages Downloaded per Second graph, you must select **Pages per second (HTML Mode only)** from the script's run-time settings Preferences tab before running your session.

Like throughput, downloaded pages per second is a representation of the amount of data that the Vusers received from the server at any given second.

- The Throughput graph takes into account each resource and its size (for example, the size of each *.gif* file, the size of each Web page).
- The Pages Downloaded per Second graph takes into account simply the number of pages.

In the following example, the Throughput graph is compared with the Pages Downloaded per Second graph. It is apparent from the graph that throughput is not proportional to the number of pages downloaded per second.

For example, between 15 and 16 seconds into the session run, the throughput decreased while the number of pages downloaded per second increased.



Retries per Second Graph

The **Retries Per Second** graph shows the number of attempted Web server connections (y-axis) as a function of the elapsed time in the session step (x-axis). A server connection is retried when the initial connection was unauthorized, when proxy authentication is required, when the initial connection was closed by the server, when the initial connection to the server could not be made, or when the server was initially unable to resolve the load generator's IP address.

Connections Graph

The **Connections** graph shows the number of open TCP/IP connections (y-axis) at each point in time of the session step (x-axis). Note that one HTML page may cause the browser to open several connections, when links on the page go to different Web addresses. Two connections are opened for each Web server.

This graph is useful in indicating when additional connections are needed. For example, if the number of connections reaches a plateau, and the transaction response time increases sharply, adding connections would probably cause a dramatic improvement in performance (reduction in the transaction response time).

Connections per Second Graph

The **Connections Per Second** graph shows the number of new TCP/IP connections (y-axis) opened each second of the session step (x-axis). This number should be a small fraction of the number of hits per second, because new TCP/IP connections are very expensive in terms of server, router and network resource consumption. Ideally, many HTTP requests should use the same connection, instead of opening a new connection for each request.

SSL Connections per Second Graph

The **SSL Connections per Second** graph shows the number of new and reused SSL Connections (y-axis) opened in each second of the session step (x-axis). An SSL connection is opened by the browser after a TCP/IP connection has been opened to a secure server.

Because creating a new SSL connection entails heavy resource consumption, you should try to open as few new SSL connections as possible; once you've established an SSL connection, you should reuse it. There should be no more than one new SSL connection per Vuser. If you've configured LoadRunner Tuning Module to simulate a new Vuser at each iteration (via the Browser Emulation tab in the Run-Time Settings menu), you should have no more than one new SSL connection per Vuser per iteration. Ideally, you should have very few new TCP/IP and SSL connections each second.

29

System Resource Monitoring

You can monitor a machine's system resource usage during a session step run using LoadRunner Tuning Module's System Resource monitors.

This chapter describes:

- Configuring the Windows Resources Monitor
- Configuring the UNIX Resources Monitor
- Configuring an rstatd Daemon on UNIX
- Configuring the SNMP Resources Monitor
- Configuring the Antara FlameThrower Monitor

About System Resource Monitoring

A primary factor in a transaction's response time is its system resource usage. Using the LoadRunner Tuning Module resource monitors, you can monitor the Windows, UNIX, SNMP, and Antara FlameThrower resources on a machine during a session step run, and determine why a bottleneck occurred on a particular machine.

The Windows measurements correspond to the built-in counters available from the Windows Performance Monitor.

The UNIX measurements include those available by the *rstatd* daemon: average load, collision rate, context switch rate, CPU utilization, incoming packets error rate, incoming packets rate, interrupt rate, outgoing packets error rate, outgoing packets rate, page-in rate, page-out rate, paging rate, swap-in rate, swap-out rate, system mode CPU utilization, and user mode CPU utilization.

Note: You must configure an *rstatd* daemon on all UNIX machines being monitored. For information on how to configure an *rstatd* daemon, refer to the UNIX *man* pages, or see "Configuring an rstatd Daemon on UNIX," on page 346.

The SNMP monitor is available for monitoring machines using the Simple Network Management Protocol (SNMP). SNMP monitoring is platform independent.

The Antara FlameThrower monitor can measure the following performance counters: Layer, TCP, HTTP, SSL/HTTPS, Sticky SLB, FTP, SMTP, POP3, DNS, and Attacks.

The resource monitors are automatically enabled when you execute a session step. However, you must specify the machine you want to monitor and which resources to monitor for each machine. You can also add or remove machines and resources during the session step run.

Configuring the Windows Resources Monitor

Windows NT and Windows 2000 measurements correspond to the built-in counters available from the Windows Performance Monitor.

Note: To monitor a Windows NT or 2000 machine through a firewall, use TCP, port 139.

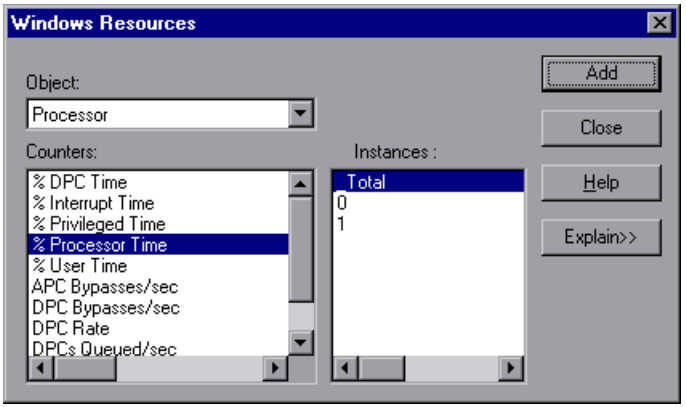
To configure the Windows Resources monitor:

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the System Resources category, select **Windows Resources**, and then click **Add**. The Windows Resources dialog box is displayed.



5 Select an object, a counter, and an instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

For a list of the available performance counters, see **page 341**.

Note: To change the default counters for the Windows machine monitor, see "Changing a Monitor's Default Counters," on page 775.

If you are monitoring a Win2000 machine, some of the NT machine default counters may not be available (such as % Total CPU usage and Interrupts/sec).

---

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: If you want to monitor a remote Windows machine that does not use Windows domain security, you must authenticate the Console machine on the remote Windows machine. To authenticate the Console machine, create an account, or change the password of the account used to log on to the Console so that it matches the password and user name used to log on to the remote monitored Windows machine. When the remote Windows machine requests another machine's resources, it sends the logged-in user name and password of the machine requesting the resources.

Windows Resources Performance Counters

The following default measurements are available for Windows machines:

Object	Measurement	Description
System	**% Total Processor Time**	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4th of the processors are 100% busy this is 25%. It can be viewed as the fraction of the time spent doing useful work. Each processor is assigned an Idle thread in the Idle process which consumes those unproductive processor cycles not used by any other threads.
System	**File Data Operations/sec**	The rate at which the computer issues read and write operations to file system devices. This does not include File Control Operations.
Processor	**% Processor Time (Windows 2000)**	The percentage of time that the processor is executing a non-idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting that value from 100%. (Each processor has an idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.

Object	Measurement	Description
System	**Processor Queue Length**	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Memory	**Page Faults/sec**	This is a count of the page faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in the main memory. A page fault will not cause the page to be fetched from disk if that page is on the standby list (and hence already in main memory), or if it is in use by another process with which the page is shared.
PhysicalDisk	**% Disk Time**	The percentage of elapsed time that the selected disk drive is busy servicing read or write requests.
Memory	**Pool Nonpaged Bytes**	The number of bytes in the nonpaged pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged pool pages cannot be paged out to the paging file. They remain in main memory as long as they are allocated.

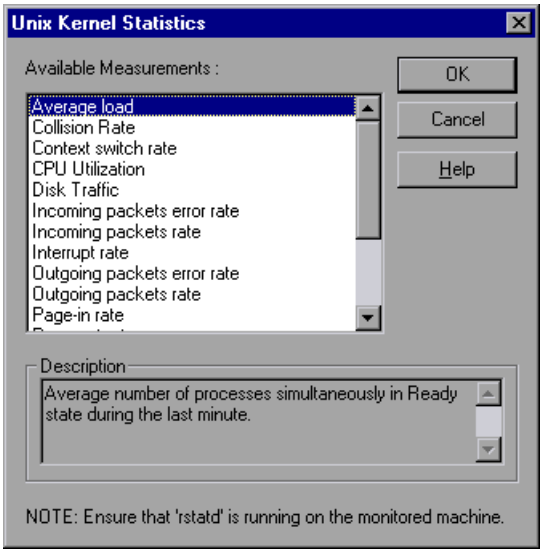
Object	Measurement	Description
Memory	**Pages/sec**	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
System	**Total Interrupts/sec**	The rate at which the computer is receiving and servicing hardware interrupts. The devices that can generate interrupts are the system timer, the mouse, data communication lines, network interface cards, and other peripheral devices. This counter provides an indication of how busy these devices are on a computer-wide basis. See also Processor:Interrupts/sec.
Objects	**Threads**	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.
Process	**Private Bytes**	The current number of bytes that the process has allocated that cannot be shared with other processes.

Configuring the UNIX Resources Monitor

The UNIX kernel statistics measurements include those available by the *rstatd* daemon: average load, collision rate, context switch rate, CPU utilization, incoming packets error rate, incoming packets rate, interrupt rate, outgoing packets error rate, outgoing packets rate, page-in rate, page-out rate, paging rate, swap-in rate, swap-out rate, system mode CPU utilization, and user mode CPU utilization.

To configure the UNIX Resources monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the System Resources category, select **UNIX Resources**, and then click **Add**. The UNIX Kernel Statistics dialog box opens, displaying the available measurements and server properties.



For a list of the available performance counters, see **page 345**.

Note: To change the default counters for the UNIX monitor, see "Changing a Monitor's Default Counters," on page 775.

---

5 To add UNIX measurements to the monitor list, select the desired measurements, and click **OK**.

6 Click **OK** in the UNIX Kernel Statistics dialog box to activate the UNIX monitor.

Note: Ensure that the rstatd daemon is correctly configured and running on the monitored UNIX machine. For more information, see "Configuring an rstatd Daemon on UNIX," on page 346.

UNIX Resources Performance Counters

The following default measurements are available for the UNIX machine:

Measurement	Description
Average load	Average number of processes simultaneously in READY state during the last minute
Collision rate	Collisions per second detected on the Ethernet
Context switches rate	Number of switches between processes or threads, per second
CPU utilization	Percent of time that the CPU is utilized
Disk rate	Rate of disk transfers
Incoming packets error rate	Errors per second while receiving Ethernet packets
Incoming packets rate	Incoming Ethernet packets per second

Measurement	Description
Interrupt rate	Number of device interrupts per second
Outgoing packets errors rate	Errors per second while sending Ethernet packets
Outgoing packets rate	Outgoing Ethernet packets per second
Page-in rate	Number of pages read to physical memory, per second
Page-out rate	Number of pages written to pagefile(s) and removed from physical memory, per second
Paging rate	Number of pages read to physical memory or written to pagefile(s), per second
Swap-in rate	Number of processes being swapped
Swap-out rate	Number of processes being swapped
System mode CPU utilization	Percent of time that the CPU is utilized in system mode
User mode CPU utilization	Percent of time that the CPU is utilized in user mode

Configuring an rstatd Daemon on UNIX

To monitor UNIX resources, you must configure the rstatd daemon. Note that the rstatd daemon might already be configured, because when a machine receives an rstatd request, the inetd on that machine activates the rstatd automatically.

To verify whether the rstatd daemon is already configured:

The *rup* command reports various machine statistics, including rstatd configuration. Run the following command to view the machine statistics:

```
>rup host
```

You can also use **lr_host_monitor** and see if it returns any relevant statistics.

If the command returns meaningful statistics, the rstatd daemon is already configured and activated. If not, or if you receive an error message, the rstatd daemon is not configured.

To configure the rstatd daemon:

1. Run the command: *su root*
2. Go to */etc/inetd.conf* and look for the rstatd row (it begins with the word rstatd). If it is commented out (with a #), remove the comment directive, and save the file.
3. From the command line, run:

   kill -1 inet_pid

   where *inet_pid* is the pid of the inetd process. This instructs the inetd to rescan the */etc/inetd.conf* file and register all daemons which are uncommented, including the rstatd daemon.
4. Run *rup* again.

   If the command still does not indicate that the rstatd daemon is configured, contact your system administrator.

Note: To monitor a UNIX machine through a firewall, you must run a UNIX utility called rpcinfo and identify the rstatd's port number. By running rpcinfo -p <hostname>, you will receive a list of all RPC servers registered in the host's portmapper, along with the port number. This list will not change until rstatd is stopped and rerun.

Some firewalls allow you to open an RPC program number instead of a port. In such cases, open program 100001. If are prompted to include a version number, specify versions 3 and 4.

Configuring the SNMP Resources Monitor

The SNMP Resources monitor is available for monitoring any machine that runs an SNMP agent, using the Simple Network Management Protocol (SNMP).

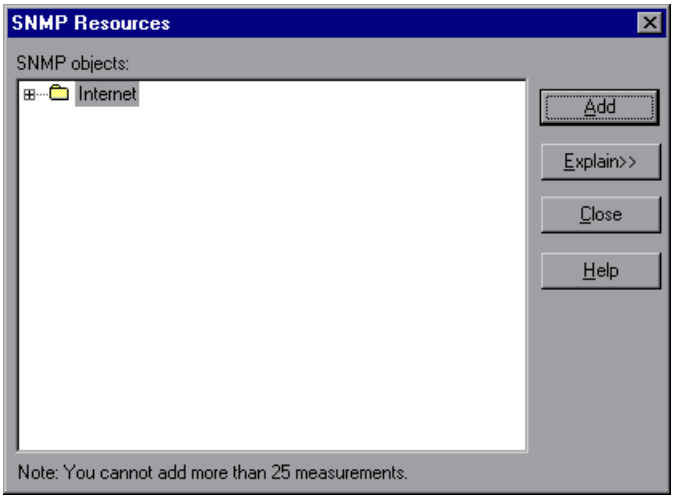
Note: You can specify a port number in the *snmp.cfg* file. If you do not specify a port, LoadRunner Tuning Module connects to default SNMP port 161. You can also specify a machine name in the following format: *<server name>:<port number>*

To monitor SNMP resources through a firewall, use ports 161 or 162.

To configure the SNMP Resources monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the System Resources category, select **SNMP**, and then click **Add**.

The SNMP Resources dialog box opens.



5 Browse the SNMP Object tree.

6 To measure an object, select it, and click **Add**. For a description of each resource, click **Explain**>> to expand the dialog box. Add all the desired resources to the list, and click **Close**.

Note: The SNMP monitor can only monitor up to 25 measurements.

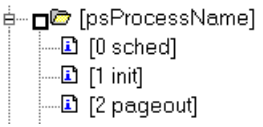
---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

You can modify the list of resources that you want to monitor at any point during the session step. Note that a session step does not have to be active in order for you to monitor the resources on a remote machine.

Note: You can improve the level of measurement information for the SNMP monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the *<LoadRunner Tuning Module root folder>\dat\monitors\snmp.cfg* file:
SNMP_show_string_nodes=1

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the SNMP Add Measurements dialog box opens, the information is reread from the *snmp.cfg* file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Configuring the Antara FlameThrower Monitor

You select measurements to monitor the Antara FlameThrower server using the Antara FlameThrower Monitor Configuration dialog box.

To configure the Antara FlameThrower monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the System Resources category, select **Antara FlameThrower**, and then click **Add**.

   The Antara FlameThrower Monitor Configuration dialog box opens.
5. Browse the Measured Components tree.
6. Check the required performance counters in the Antara FlameThrower Monitor Configuration window's right pane.

   For a list of the available performance counters, see **page 352**.
7. Click **OK** in the Antara FlameThrower Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the Antara FlameThrower monitor.

Antara FlameThrower Performance Counters

The following tables describe the counters that can be monitored:

Layer Performance Counters

Measurement	Description
TxBytes	The total number of Layer 2 data bytes transmitted.
TxByteRate(/sec)	The number of Layer 2 data bytes transmitted per second.
TxFrames	The total number of packets transmitted.
TxFrameRate(/sec)	The number of packets transmitted per second.
RxBytes	The total number of Layer 2 data bytes received.
RxByteRate(/sec)	The number of Layer 2 data bytes received per second.
RxFrames	The total number of packets received.
RxFrameRate(/sec)	The number of packets received per second.

TCP Performance Counters

Measurement	Description
ActiveTCPConns	Total number of currently active TCP connections.
SuccTCPConns	Total number of SYN ACK packets received.
SuccTCPConn Rate(/sec)	Number of SYN ACK packets received per second.
TCPConnLatency(milisec)	Interval between transmitting a SYN packet and receiving a SYN ACK reply packet in msec.
MinTCPConn Latency(milisec)	Minimum TCPConnectionLatency in msec.
MaxTCPConn Latency(milisec)	Maximum TCPConnectionLatency in msec.
TCPSndConnClose	Total number of FIN or FIN ACK packets transmitted (Client).

Measurement	Description
TCPRcvConnClose	Total number of FIN or FIN ACK packets received (Client).
TCPSndResets	Total number of RST packets transmitted.
TCPRcvResets	Total number of RST packets received.
SYNSent	Total number of SYN packets transmitted.
SYNSentRate(/sec)	Number of SYN packets transmitted per second.
SYNAckSent	Total number of SYN ACK packets transmitted.
SYNAckRate(/sec)	Number of SYN ACK packets transmitted per second.

HTTP Performance Counters

Measurement	Description
HTTPRequests	Total number of HTTP Request command packets transmitted.
HTTPRequestRate (/sec)	Number of HTTP Request packets transmitted per second.
AvgHTTPData Latency(milisecs)	The average HTTP Data Latency over the past second in msec.
HTTPData Latency(milisecs)	Interval between transmitting a Request packet and receiving a response in msec.
DataThroughput (bytes/sec)	The number of data bytes received from the HTTP server per second.
MinHTTPData Latency(milisecs)	Minimum HTTPDataLatency in msec.
MaxHTTPData Latency(milisecs)	Maximum HTTPDataLatency in msec.
MinData Throughput (bytes/sec)	Minimum HTTPDataThroughput in seconds.

Measurement	Description
MaxData Throughput (bytes/sec)	Maximum HTTPDataThroughput in seconds.
SuccHTTPRequests	Total number of successful HTTP Request Replies (200 OK) received.
SuccHTTPRequest Rate(/sec)	Number of successful HTTP Request Replies (200 OK) received per second.
UnSuccHTTP Requests	Number of unsuccessful HTTP Requests.

SSL/HTTPS Performance Counters

Measurement	Description
SSLConnections	Number of ClientHello messages sent by the Client.
SSLConnection Rate(/sec)	Number of ClientHello messages sent per second.
SuccSSL Connections	Number of successful SSL Connections. A successful connection is one in which the Client receives the Server's finished handshake message without any errors.
SuccSSLConnectionRate(/sec)	Number of successful SSL connections established per second.
SSLAlertErrors	Number of SSL alert messages received by the client (e.g. bad_record_mac, decryption_failed, handshake_failure, etc..)
SuccSSLResumed Sessions	Number of SSL Sessions that were successfully resumed.
FailedSSLResumed Sessions	Number of SSL Sessions that were unable to be resumed.

Sticky SLB Performance Counters

Measurement	Description
Cookie AuthenticationFail	The number of Cookie's that were not authenticated by the Server.
SuccCookie Authentication	The number of Cookie's authenticated by the server.
SSLClientHellos	The number of Client Hello packets sent to the server.
SSLServerHellos	The number of Server Hello packets sent to back to the client.
SSLSessionsFailed	The number of Session ID's that were not authenticated by the server.
SSLSessions Resumed	The number of Session ID's authenticated by the server.
succSSLClientHellos	The number of Client Hello replies received by the client or packets received by the server.
succSSLServerHellos	The number of Server Hello's received by the client.

FTP Performance Counters

Measurement	Description
TPUsers	Total number of Ftp User command packets transmitted.
FTPUserRate(/sec)	Number of Ftp User command packets transmitted per second.
FTPUserLatency (milisecs)	Interval between transmitting a Ftp User command packet and receiving a response in msec.
MinFTPUserLatency(milisecs)	Minimum FTPUsersLatency in msec.
MaxFTPUserLatency(milisecs)	Maximum FTPUsersLatency in msec.
SuccFTPUsers	Total number of successful Ftp User command replies received.

Measurement	Description
SuccFTPUserRate (/sec)	Number of successful Ftp User command replies received per second.
FTPPasses	Total number of FTP PASS packets transmitted.
FTPPassRate(/sec)	Number of FTP PASS packets transmitted per second.
FTPPassLatency (milisecs)	Interval between transmitting a Ftp PASS packet and receiving a response in msec.
MinFTPPassLatency (milisecs)	Minimum FTPPassLatency in msec.
MaxFTPPassLatency(milisecs)	Maximum FTPPassLatency in msec.
SuccFTPPasses	Total number of successful FTP PASS replies received.
SuccFTPPassRate (/sec)	Number of successful FTP PASS replies received per second.
FTPControl Connections	Total number of SYN packets transmitted by the FTP client.
FTPControl ConnectionRate (/sec)	Number of SYN packets transmitted by the FTP client per second.
SuccFTPControl Connections	Total number of SYN ACK packets received by the FTP client.
SuccFTPControl ConnectionRate (/sec)	Number of SYN ACK packets received by the FTP Client per second.
FTPData Connections	Number of SYN ACK packets received by the FTP client per second.
FTPDataConnectionR ate(/sec)	Number of SYN ACK packets transmitted by the FTP Client or received by the FTP Server per second.
SuccFTPData Connections	Total number of SYN ACK packets transmitted by the FTP Client or received by the FTP Server.

Measurement	Description
SuccFTPData ConnectionRate (/sec)	Number of SYN ACK packets received by the FTP server per second.
FtpAuthFailed	Total number of error replies received by the FTP client.
FTPGets	Total number of client Get requests.
FTPPuts	Total number of client Put requests.
SuccFTPGets	Total number of successful Get requests (data has been successfully transferred from server to client).
SuccFTPPuts	Total number of successful Put requests (data has been successfully transferred from client to server) .

SMTP Performance Counters

Measurement	Description
SMTPHelos	Total number of HELO packets transmitted.
SMTPHeloRate(/sec)	Number of HELO packets transmitted per second.
SMTPHeloLatency (milisecs)	Interval between transmitting a HELO packet and receiving a response in msec.
MinSMTPHelo Latency(milisecs)	Minimum SMTPHeloLatency in msec.
MaxSMTPHelo Latency(milisecs)	Maximum SMTPHeloLatency in msec.
SuccSMTPHelos	Total number of successful HELO replies received.
SuccSMTPHelo Rate(/sec)	Number of successful HELO replies received per second.
SMTPMailFroms	Total number of Mail From packets transmitted.
SMTPMailFromRate(/sec)	Number of Mail From packets transmitted per second.
SMTPMailFrom Latency(milisecs)	Interval between transmitting a Mail From packet and receiving a response in msec.

Measurement	Description
MinSMTPMailFromLatency(milisecs)	Minimum SMTPMailFromLatency in msec.
MaxSMTPMailFromLatency(milisecs)	Maximum SMTPMailFromLatency in msec.
SuccSMTPMail Froms	Total number of successful Mail From replies received.
SuccSMTPMailFromRate(/sec)	Number of successful Mail From replies received per second.
SMTPRcptTos	Total number of RcptTo packets transmitted.
SMTPRcptToRate (/sec)	Number of RcptTo packets transmitted per second.
SMTPRcptTo Latency(milisecs)	Interval between transmitting a RcptTo packet and receiving a response in msec.
MinSMTPRcptTo Latency(milisecs)	Minimum SMTPRcptToLatency in msec.
MaxSMTPRcptTo Latency(milisecs)	Maximum SMTPRcptToLatency in msec.
SuccSMTPRcptTos	Total number of successful RcptTo replies received.
SuccSMTPRcptTo Rate(/sec)	Number of successful RcptTo replies received per second.
SMTPDatas	Total number of Data packets transmitted.
SMTPDataRate(/sec)	Number of Data packets transmitted per second.
SMTPDataLatency (milisecs)	Interval between transmitting a Data packet and receiving a response in msec.
MinSMTPData Latency(milisecs)	Minimum SMTPDataLatency in msec.
MaxSMTPData Latency(milisecs)	Maximum SMTPDataLatency in msec.

Measurement	Description
SuccSMTPDatas	Total number of successful Data replies received.
SuccSMTPDataRate (/sec)	Number of successful Data replies received per second.

POP3 Performance Counters

Measurement	Description
POP3Users	Total number of Pop3 User command packets transmitted.
POP3UserRate(/sec)	Number of Pop3 User command packets transmitted per second.
POP3UserLatency (milisecs)	Interval between transmitting a Pop3 User command packet and receiving a response in msec.
MinPOP3User Latency(milisecs)	Minimum POP3UserLatency in msec.
MaxPOP3User Latency(milisecs)	Maximum POP3UserLatency in msec.
SuccPOP3Users	Total number of successful Pop3 User replies received.
SuccPOP3UserRate (/sec)	Number of successful Pop3 User replies received per second.
POP3Passes	Total number of Pop3 Pass command packets transmitted.
POP3PassRate(/sec)	Number of Pop3 Pass command packets transmitted per second.
POP3PassLatency (milisecs)	Interval between transmitting a Pop3 Pass packet and receiving a response in msec.
MinPOP3Pass Latency(milisecs)	Minimum POP3PassLatency in msec.
MaxPOP3Pass Latency(milisecs)	Maximum POP3PassLatency in msec.
SuccPOP3Passes	Total number of successful Pop3 Pass replies received.

Measurement	Description
SuccPOP3PassRate (/sec)	Number of successful Pop3 Pass replies received per second.
POP3Stats	Total number of Pop3 Stat command packets sent.
POP3StatRate(/sec)	Number of Pop3 Stat command packets transmitted per second.
POP3StatLatency (milisecs)	Interval between transmitting a Pop3 Stat packet and receiving a response in msec.
MinPOP3Stat Latency(milisecs)	Minimum POP3StartLatency in msec.
MaxPOP3Stat Latency(milisecs)	Maximum POP3StartLatency in msec.
SuccPOP3Stats	Total number of successful Pop3 Stat replies received.
SuccPOP3StatRate (/sec)	Number of successful Pop3 Stat replies received per second.
POP3Lists	Total number of Pop3 List command packets transmitted.
POP3ListRate(/sec)	Number of Pop3 List command packets transmitted per second.
POP3ListLatency (milisecs)	Interval between transmitting a Pop3 List packet and receiving a response in msec.
MinPOP3List Latency(milisecs)	Minimum POP3ListLatency in msec.
MaxPOP3List Latency(milisecs)	Maximum POP3ListLatency in msec.
SuccPOP3Lists	Total number of successful Pop3Lists received.
SuccPOP3ListRate (/sec)	Number of successful Pop3Lists received per second.
POP3Retrs	Total number of Pop3 Retr packets transmitted.
POP3RetrRate(/sec)	Number of Pop3 Retr packets transmitted per second.
POP3RetrLatency (milisecs)	Interval between transmitting a Pop3 Retr packet and receiving a response in msec.

Measurement	Description
MinPOP3Retr Latency(milisecs)	Minimum POP3RetrLatency in msec.
MaxPOP3Retr Latency(milisecs)	Maximum POP3RetrLatency in msec.
SuccPOP3Retrs	Total number of successful Pop3Retrs received.
SuccPOP3RetrRate (/sec)	Number of successful Pop3Retrs received per second.

DNS Performance Counters

Measurement	Description
SuccPrimaryDNS Request	Total number of Successful DNS requests made to the Primary DNS server.
SuccSecondaryDNSR equest	Total number of Successful DNS requests made to the Secondary DNS server.
SuccDNSData RequestRate(/sec)	Number of Successful DNS Request packets transmitted per second.
PrimaryDNSFailure	Total number of DNS requests failures received from the Primary DNS server.
PrimaryDNSRequest	Total number of DNS requests made to the Primary DNS server.
SecondaryDNS Failure	Total number of DNS requests failures received from the Secondary DNS server.
SecondaryDNS Request	Total number of DNS requests made to the Secondary DNS server.
MinDNSData Latency	Minimum DNS Data Latency in msec.
MaxDNSData Latency	Maximum DNS Data Latency in msec.
CurDNSData Latency	Interval between sending a DNS request packet and receiving a response in msec.

Measurement	Description
DNSDataRequest Rate(/sec)	Number of DNS Request packets transmitted per second.
NoOf ReTransmission	Total number of DNS Request packets re
NoOfAnswers	Total number of Answers to the DNS Request packets.

Attacks Performance Counters

Measurement	Description
Attacks	Total number of attack packets transmitted (All Attacks)
AttackRate(/sec)	Number of attack packets transmitted per second (ARP, Land, Ping, SYN, and Smurf)
Havoc Flood	Number of Havoc packets generated (Stacheldraht only)
Icmp Flood	Number of ICMP attack packets generated (TFN, TFN2K, & Stacheldraht)
Mix Flood	Number of Mix packets generated (TFN2K only)
Mstream Flood	Number of Mstream packets generated (Stacheldraht only)
Null Flood	Number of Null packets generated (Stacheldraht only)
Smurf Flood	Number of Smurf packets generated (TFN, TFN2K, & Stacheldraht)
Syn Flood	Number of SYN packets generated (TFN, TFN2K, & Stacheldraht)
Targa Flood	Number of Targa packets generated (TFN2K only)
Udp Flood	Number of UDP packets generated (All DDoS Attacks only)

30

Network Monitoring

You can use Network monitoring to determine whether your network is causing a delay in the session step. You can also determine the problematic network segment.

Note: You must have administrator privileges on the Windows source machine in order to run the Network monitor (unless you are using the ICMP protocol).

This chapter describes:

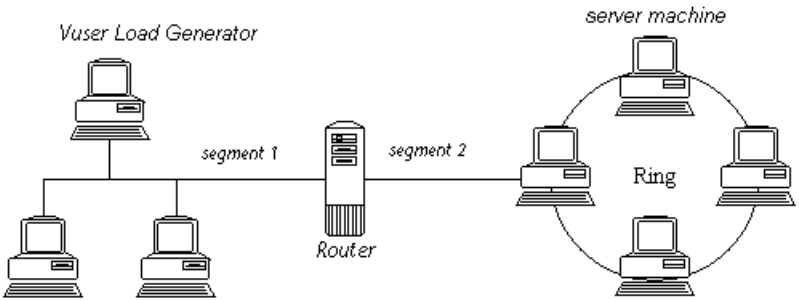
- Network Monitoring from a UNIX Source Machine
- Configuring the Network Delay Time Monitor
- Viewing the Network Delay Time Graph

About Network Monitoring

Network configuration is a primary factor in the performance of applications. A poorly designed network can slow client activity to unacceptable levels.

In a true Web or client/server system, there are many network segments. A single network segment with poor performance can affect the entire system.

The following diagram shows a typical network. In order to go from the server machine to the Vuser machine, data must travel over several segments.



To measure network performance, the Network monitor sends packets of data across the network. When a packet returns, the monitor calculates the time it takes for the packet to go to the requested node and return. This time is the delay which appears in the Network Delay Time graph.

Using the online Network Delay Time graph, you can locate the network-related problem so that it can be fixed.

Note: The delays from the source machine to each of the nodes are measured concurrently, yet independently. It is therefore possible that the delay from the source machine to one of the nodes could be greater than the delay for the complete path between the source and destination machines.

Network Monitoring from a UNIX Source Machine

You can run the Network monitor on UNIX machines, using UDP or ICMP.

Note: If the server is running HP-UX, you need to use ICMP.

Before running the Network monitor from a UNIX source machine:

- configure the source machine by assigning root permissions to the *merc_webtrace* process.
- make the necessary adjustments to either connect to the source machine through RSH, or through the agent.

Configuring the Source Machine

To configure the source machine, where LoadRunner Tuning Module is installed locally:

To assign root permissions to the *merc_webtrace* process, add an s-bit to *merc_webtrace*'s permissions, as follows:

1. Log in to the source machine as root.
2. Type: cd <LoadRunner_Tuning_Module_installation>/bin to change to the *bin* directory.
3. Type: chown root merc_webtrace to make the root user the owner of the *merc_webtrace* file.
4. Type: chmod +s merc_webtrace to add the s-bit to the file permissions.
5. To verify, type ls -l merc_webtrace. The permissions should look like: -rwsrwsr-x.

To configure the source machine, where LoadRunner Tuning Module is installed on the network:

In a LoadRunner Tuning Module network installation, the *merc_webtrace* process is on the network, not on the source machine disk. The following procedure copies the *merc_webtrace* file to the local disk, configures *mdrv.dat* to recognize the process, and assigns root permissions to *merc_webtrace*:

1 Copy *merc_webtrace* from *<LoadRunner Tuning Module_installation>/bin* to anywhere on the local disk of the source machine. For example, to copy the file to the */local/LoadRunner Tuning Module* directory, type: cp /net/tools/LoadRunner_Tuning_Module_installation/bin/merc_webtrace /local/LoadRunner_Tuning_Module

---

Note: All of the source machines that use the same network installation must copy *merc_webtrace* to the identical directory path on their local disk (for example, */local/LoadRunner Tuning Module*), since all of them use the same *mdrv.dat*.

---

2 Add the following line to the *<LoadRunner_Tuning_Module_installation>/dat/mdrv.dat* file, in the [monitors_server] section:

```
ExtCmdLine=-merc_webtrace_path /local/xxx
```

3 Log in to the source machine as root.

4 Type: cd LoadRunner_Tuning_Module_installation/bin to change to the *bin* directory.

5 Type: chown root merc_webtrace to make the root user the owner of the *merc_webtrace* file.

6 Type: chmod +s merc_webtrace to add the s-bit to the file permissions.

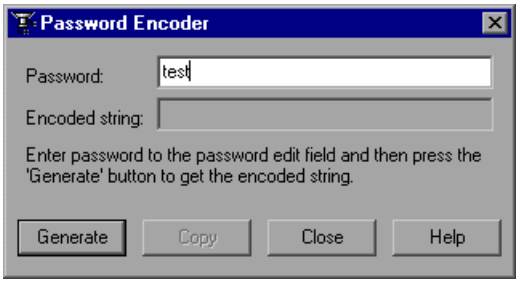
7 To verify, type ls -l merc_webtrace. The permissions should look like: -rwsrwsr-x.

Connecting to the Source Machine Through RSH

If the Console is connected to the source machine through RSH (default connection mode), then you don't need to activate the agent daemon. Before running the Network monitor the first time, you enter an encrypted user name and password in the Network monitor configuration file.

To create an encrypted user name and password:

1 On the Windows taskbar, click **Start**, point to **Programs** > **LoadRunner Tuning Module** > **Tools**, and click **Password Encoder**. The Password Encoder window opens.



2 In the **Password** box, type your RSH user name and password, separated by a vertical bar symbol. For example, myname|mypw.

3 Click **Generate**. An encoded string is displayed in the Encoded string field.

4 Click **Copy** to copy the encoded string to the clipboard.

5 Add the following line to the *<LoadRunner_Tuning_Module_installation>/dat/monitors/ndm.cfg* file, in the [hosts] section:

```
Host = <encrypted string copied from clipboard>
```

6 Close and open the current session step. LoadRunner Tuning Module will read the updated configuration file and recognize the source machine for monitoring.

Connecting to the Source Machine Through the Agent

If the Console is not connected to the source machine through RSH, then make sure that the agent daemon is active on the source machine before running the Network monitor. For more information about working without RSH, refer to the section titled "UNIX Shell" in Appendix B, "Troubleshooting the Console."

To activate the agent daemon:

If you are not working in RSH, invoke the agent daemon on the source machine.

1. Type m_daemon_setup -install from the *<LoadRunner_Tuning_Module_installation>/bin* directory.
2. Make sure that the agent daemon is running whenever you activate the Network monitor.

Tip: To deactivate the agent daemon, type m_daemon_setup -remove.

Configuring the Network Delay Time Monitor

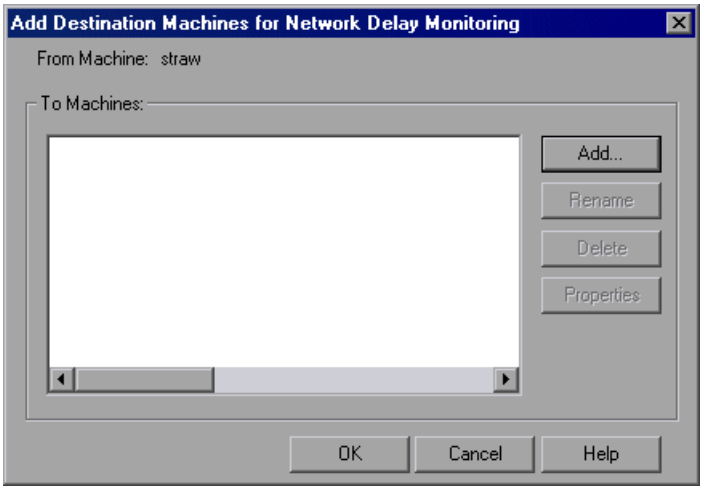
You configure the Network monitor from the topology window of the Console before you begin running a session step. Using the Add Destination Machines for Network Delay Monitoring dialog box, you select the network path you want to monitor.

Note: To enable network monitoring, you must install the LoadRunner agent on the source machine. You do not have to install the LoadRunner agent on the destination machine.

To configure the Network monitor:

1 Click **Monitors** to display the Monitors Configuration dialog box, click **Add Monitor**, and check **Show All Available Monitors**.

2 In the list of monitors in the left section, expand the Load Generator category, click **Network Delay**, and then click **Add**. The Add Destination Machines for Network Delay Monitoring dialog box is displayed.



3 To add a machine, click **Add**, enter the name of the destination machine, and click **OK**. The name of the machine appears in the Add Destination Machines for Network Delay Monitoring dialog box. Repeat this procedure for each path you want to monitor.

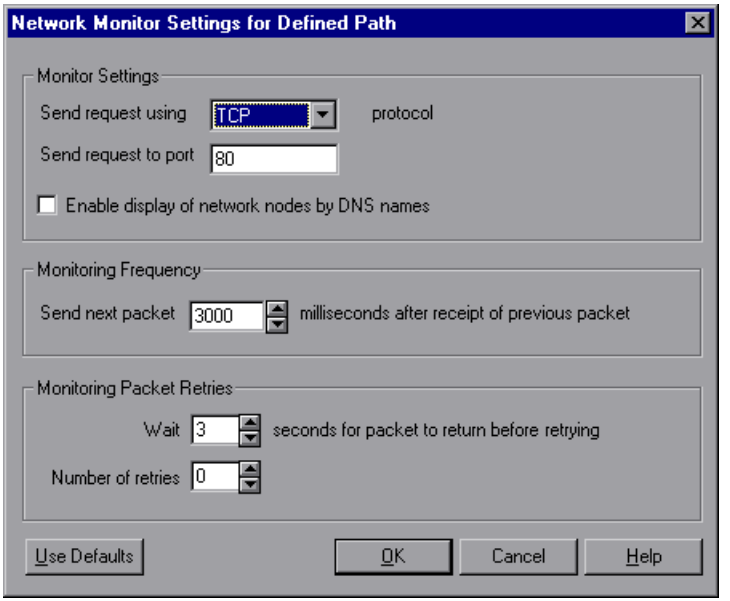
Note: If the destination machine is *localhost*, enter the local machine's name and not *localhost*.

---

4 To rename a machine, click **Rename**, and enter a new name for the machine.

5 To delete a machine, select it and click **Delete**.

6 Click **Properties** to configure additional network monitor settings. The Network Monitor Settings for Defined Path dialog box opens.



7 In the Monitor Settings box, select the protocol and enter the port number being used by the network path. The Network monitor supports three protocols: TCP, UDP, and ICMP. It is recommended that you use the default protocol. In Windows, the default is TCP, and in UNIX, the default is UDP.

8 Select **Enable display of network nodes by DNS names** if you want to view the DNS name of each node along the network path, in addition to its IP address. Note that selecting this option will decrease the speed of the Network monitor.

9 In the Monitoring Frequency box, select the number of milliseconds the monitor should wait between receiving a packet and sending out the next packet. The default value is 3000 milliseconds. If you have a long, steady session step, you can increase the interval by several seconds.

10 In the Monitoring Packet Retries box, select the maximum number of seconds that the monitor should wait for a packet to return before it retries to send the packet. The default value is 3 seconds. If your network is very large and loaded (an internet connection with a low capacity), you should increase the value by several seconds. If you have a small network (such as a LAN), you can decrease the value.

In addition, select the number of times the Network monitor should try resending a packet to a node if the packet is not initially returned. The default value is 0.

Network Monitoring over a Firewall

If you are monitoring a network in which there are firewalls between the source and the destination machines, you must configure the firewalls to allow the network data packets to reach their destinations.

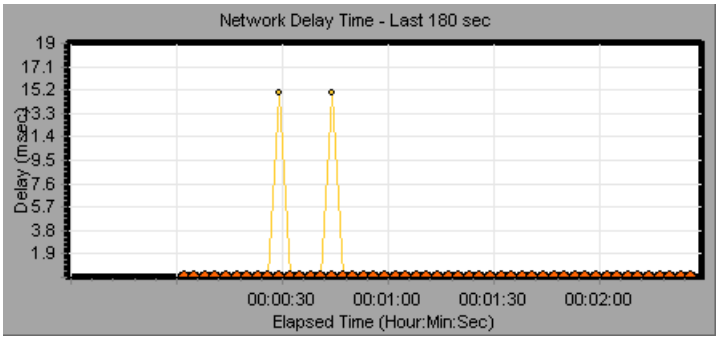
- If you are using the TCP protocol, the firewall that protects the destination machine should not block outgoing ICMP_TIMEEXCEEDED packets (packets that are sent outside the firewall from the machine). In addition, the firewall protecting the source machine should allow ICMP_TIMEEXCEEDED packets to enter, as well as TCP packets to exit.
- If you are using the ICMP protocol, the destination machine's firewall should not block incoming ICMP_ECHO_REQUEST packets, or outgoing ICMP_ECHO_REPLY and ICMP_ECHO_TIMEEXCEEDED packets. In addition, the firewall protecting the source machine should allow ICMP_ECHO_REPLY and ICMP_ECHO_TIMEEXCEEDED packets to enter, and ICMP_ECHO_REQUEST packets to exit.
- If you are using the UDP protocol, ensure that the UDP protocol can access the destination machine from the source machine. The destination machine's firewall should not block outgoing ICMP_DEST_UNREACHABLE and ICMP_ECHO_TIMEEXCEEDED packets. In addition, the firewall protecting the source machine should allow ICMP_DEST_UNREACHABLE and ICMP_ECHO_TIMEEXCEEDED packets to enter.

Note: To run the Network Delay Monitor when there are firewalls between the Console machine and the source machine, you must configure the LoadRunner Tuning Module agent, MI Listener, and Network monitor for monitoring over a firewall. For more information see "Configuring LoadRunner Agents Inside the Firewall," on page 262, "Installing and Configuring the MI Listener Outside the Firewall," on page 272, and "Configuring the Network Delay Monitor over a Firewall," on page 285.

Viewing the Network Delay Time Graph

The **Network Delay Time** graph shows the delay for the complete path between the source and destination machines (y-axis) as a function of the elapsed session step time (x-axis).

Each path defined in the Add Destination Machines for Network Delay Monitoring dialog box is represented by a separate line with a different color in the graph.



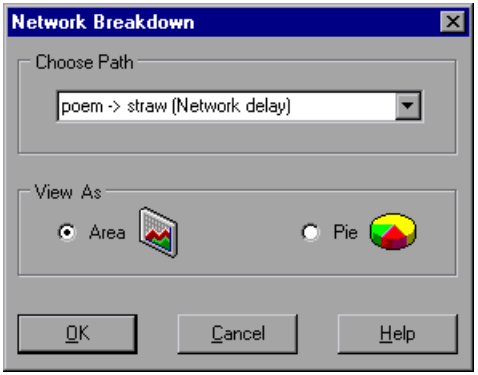
To view the DNS names of the measurements displayed in the legend, right-click the graph and select **View as DNS Name**.

To view the delay time from the source machine to each of the nodes along the network path, right-click the graph and select **Configure**. In the Graph Configuration dialog box, click **SubPaths**.

In addition, you can view the delay time for each segment of the path.

To view the delay time for the network segments:

1 Right-click the Network Delay Time graph, and select **View Segments**. The Network Breakdown dialog box opens.



2 Select the path that you want to break down.

3 Choose whether you want to view the network segments of the graph of the graph you chose as an area graph or a pie graph.

4 Click **OK** to close the Network Breakdown dialog box. The delay time for the network segments of the path you chose is displayed in the graph view area.

Note: The segment delays are measured approximately, and do not add up to the network path delay which is measured exactly. The delay for each segment of the path is estimated by calculating the delay from the source machine to one node and subtracting the delay from the source machine to another node. For example, the delay for segment B to C is calculated by measuring the delay from the source machine to point C, and subtracting the delay from the source machine to point B.

To return to the complete path delay time view, select **Hide Segments** from the right-click menu.

31

Firewall Server Performance Monitoring

During a session step run, you can monitor the firewall server in order to isolate server performance bottlenecks.

This chapter describes:

➤ Configuring the Check Point FireWall-1 Server Monitor

About the Firewall Server Monitor

The Firewall server online monitor measures the performance of a Firewall server during session step execution. In order to obtain performance data, you need to activate the Firewall server monitor before executing the session step, and indicate which statistics and measurements you want to monitor.

Configuring the Check Point FireWall-1 Server Monitor

To monitor the CheckPoint FireWall-1 server, you must select the counters you want the CheckPoint FireWall-1 server monitor to measure. You select these counters using the CheckPoint FireWall-1 SNMP Resources dialog box.

Note: You can specify a port number in the *snmp.cfg* file. If you do not specify a port number, LoadRunner Tuning Module connects to port 260, the default port for the CheckPoint FireWall-1 SNMP agent.

To configure the CheckPoint FireWall-1 server monitor:

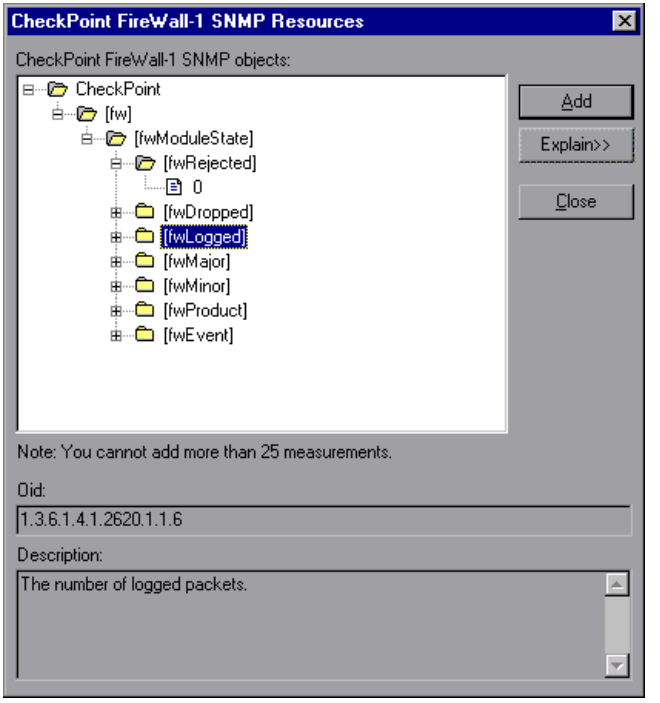
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Firewall category, select **CheckPoint FireWall-1**, and then click **Add**.

The CheckPoint FireWall-1 SNMP Resources dialog box opens.



5 Select the measurements you want to monitor.

For a list of the available performance counters, see page 378.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

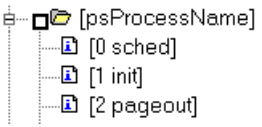
Note: The CheckPoint FireWall-1 monitor can only monitor up to 25 measurements.

---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: You can improve the level of measurement information for the CheckPoint FireWall-1 monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the *<LoadRunner Tuning Module root folder>\dat\monitors\snmp.cfg* file:
SNMP_show_string_nodes=1

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the CheckPoint FireWall-1 SNMP Resources dialog box opens, the information is reread from the *snmp.cfg* file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Check Point FireWall-1 Performance Counters

The following default counters can be monitored:

Measurement	Description
fwRejected	The number of rejected packets.
fwDropped	The number of dropped packets.
fwLogged	The number of logged packets.

32

Web Server Resource Monitoring

Using LoadRunner Tuning Module's Web Server Resource monitors, you can monitor the Apache, Microsoft IIS, iPlanet (SNMP), and iPlanet/Netscape servers during a session step run and isolate server performance bottlenecks.

This chapter describes:

- Configuring the Apache Monitor
- Configuring the Microsoft IIS Monitor
- Configuring the iPlanet/Netscape Monitor
- Configuring the iPlanet (SNMP) Monitor
- Monitoring Using a Proxy Server

About Web Server Resource Monitors

Web Server Resource monitors provide you with information about the resource usage of the Apache, Microsoft IIS, iPlanet (SNMP), and iPlanet/Netscape Web servers during session step execution. In order to obtain this data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the session step.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Note: Certain measurements or counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a Web server. For more information about these counters, see "Useful Counters for Stress Testing" on page 776.

Configuring the Apache Monitor

To monitor an Apache server you need to know the server statistics information URL. A simple way to verify the statistics information URL is to try to view it through the browser.

The URL should be in the following format:

http://*<server name/IP address>*:*<port number>*/server-status?auto

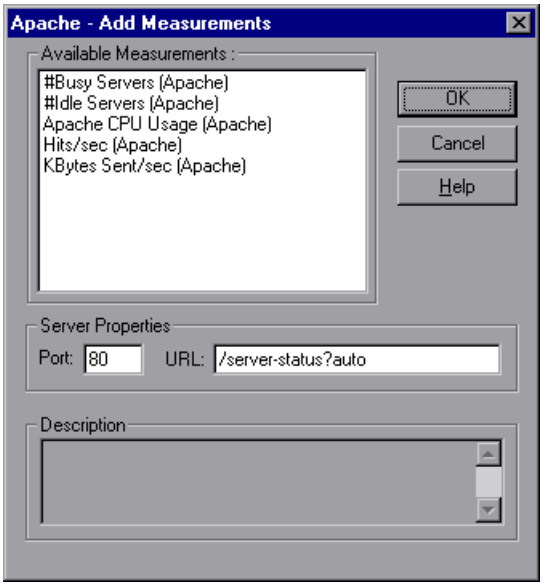
For example:

http://stimpy:80/server-status?auto

To configure the Apache monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Web Server category, select **Apache**, and then click **Add**.

The Apache - Add Measurements dialog box opens, displaying the available measurements and server properties.



5 Select the required measurements. You can select multiple measurements using the **Ctrl** key.

For a list of the available performance counters, see page 382.

6 In the Server Properties section, enter the Port number and URL (without the server name), and click **OK**. The default URL is /server-status?auto.

7 Click **OK** in the Apache dialog box to activate the monitor.

Note: The default port number and URL can vary from one server to another. Please consult your Web server administrator.

Apache Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
# Busy Servers	The number of servers in the Busy state
# Idle Servers	The number of servers in the Idle state
Apache CPU Usage	The percentage of time the CPU is utilized by the Apache server
Hits/sec	The HTTP request rate
KBytes Sent/sec	The rate at which data bytes are sent from the Web server

To change the default server properties:

1 Open the *apache.cfg* file in the *<LoadRunner Tuning Module root folder>\dat\monitors* directory.

2 Edit the following parameters after the Delimiter=: statement:

InfoURL server statistics information URL

ServerPort server port number

SamplingRate rate (milliseconds) at which the LoadRunner Tuning Module monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner Tuning Module will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor an Apache server through a firewall, use the Web server port (by default, port 80).

Configuring the Microsoft IIS Monitor

You select measurements for the Microsoft IIS Server monitor using the MS IIS dialog box.

Note: To monitor an IIS server through a firewall, use TCP, port 139.

To configure the IIS server monitor:

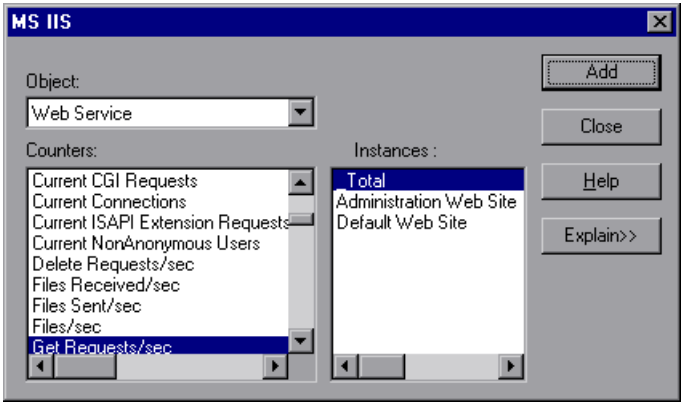
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **MS IIS**, and then click **Add**.

A dialog box displaying the Web Service object, its counters, and instances opens.



For a list of the available performance counters, see page 384.

Note: To change the default counters for the Microsoft IIS Server monitor, see "Changing a Monitor's Default Counters" on page 775.

---

5 Select a counter and an instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Microsoft IIS Performance Counters

The following table describes the default measurements that can be monitored:

Object	Measurement	Description
Web Service	**Bytes Sent/sec**	The rate at which the data bytes are sent by the Web service
Web Service	**Bytes Received/sec**	The rate at which the data bytes are received by the Web service
Web Service	**Get Requests/sec**	The rate at which HTTP requests using the GET method are made. Get requests are generally used for basic file retrievals or image maps, though they can be used with forms.
Web Service	**Post Requests/sec**	The rate at which HTTP requests using the POST method are made. Post requests are generally used for forms or gateway requests.
Web Service	**Maximum Connections**	The maximum number of simultaneous connections established with the Web service

Object	Measurement	Description
Web Service	**Current Connections**	The current number of connections established with the Web service
Web Service	**Current NonAnonymous Users**	The number of users that currently have a non-anonymous connection using the Web service
Web Service	**Not Found Errors/sec**	The rate of errors due to requests that could not be satisfied by the server because the requested document could not be found. These are generally reported to the client as an HTTP 404 error code.
Process	**Private Bytes**	The current number of bytes that the process has allocated that cannot be shared with other processes.

Configuring the iPlanet/Netscape Monitor

To monitor an iPlanet/Netscape server, you need to know the administration server URL. A simple way to verify the administration server URL, is to try to view it through the browser.

The URL should be in the following format:

http://<*admin_srv_name/IP address*>:<*port number*>/https-<*admin_srv_name/IP address*>/bin/sitemon?doit

for example:

http://lazarus:12000/https-lazarus.mercury.co.il/bin/sitemon?doit

---

Note: In some server configurations, the URL must contain the administration server name and not the IP address.

In addition, the administration server name may differ from the iPlanet/Netscape server name.

To activate the iPlanet/Netscape monitor from the Console:

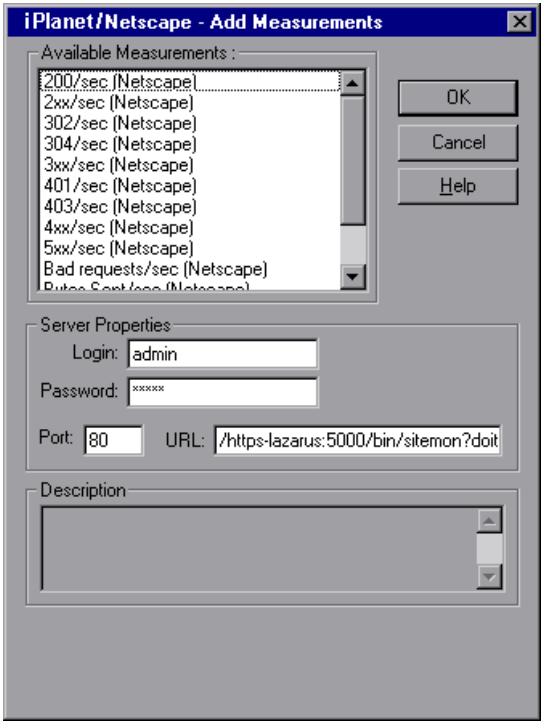
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **iPlanet/Netscape**, and then click **Add**.

The iPlanet/Netscape - Add Measurements dialog box opens, displaying the available measurements and server properties:



5 Select the required measurements. You can select multiple measurements using the **Ctrl** key.

For a list of the available performance counters, see page 387.

6 Fill in the Server Properties:

➤ Enter the user login name and password. The user must have administrator permissions on the server.

➤ Enter the port number and URL (without the server name), and click **OK**. The default URL is /https-<admin_server>/bin/sitemon?doit.

7 Click **OK** in the iPlanet/Netscape dialog box to activate the monitor.

Note: The default port number and URL can vary from one server to another. Please consult the Web server administrator. In some server configurations, the URL must contain the administration server name and not the IP address.

---

iPlanet/Netscape Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
200/sec	The rate of successful transactions being processed by the server
2xx/sec	The rate at which the server handles status codes in the 200 to 299 range
302/sec	The rate of relocated URLs being processed by the server
304/sec	The rate of requests for which the server tells the user to use a local copy of a URL instead of retrieving a newer version from the server
3xx/sec	The rate at which the server handles status codes in the 300 to 399 range
401/sec	The rate of unauthorized requests handled by the server
403/sec	The rate of forbidden URL status codes handled by the server

Measurement	Description
4xx/sec	The rate at which the server handles status codes in the 400 to 499 range
5xx/sec	The rate at which the server handles status codes 500 and higher
Bad requests/sec	The rate at which the server handles bad requests
Bytes sent/sec	The rate at which bytes of data are sent from the Web server
Hits/sec	The HTTP request rate
xxx/sec	The rate of all status codes (2xx-5xx) handled by the server, excluding timeouts and other errors that did return an HTTP status code

To change the default server properties:

1 Open the *Netscape.cfg* file in the *<LoadRunner Tuning Module root folder>\dat\monitors* directory.

2 Edit the following parameters in the [Netscape] section:

Counters number of counters that the LoadRunner Tuning Module iPlanet/Netscape monitor will show you. This value should match the number of counters defined in the file.

InfoURL server statistics information URL

ServerPort server port number

ServerLogin login name to the server

ServerPassword login password for the login name

SamplingRate rate (milliseconds) at which the LoadRunner Tuning Module monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner Tuning Module will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor an iPlanet/Netscape server through a firewall, use the iPlanet/Netscape Administration server port. Configure this port during the server installation process.

Configuring the iPlanet (SNMP) Monitor

The iPlanet (SNMP) monitor uses the Simple Network Management Protocol (SNMP) to retrieve iPlanet (SNMP) server statistics. You define the measurements for the iPlanet (SNMP) monitor using the iPlanet (SNMP) dialog box.

Note: To monitor a iPlanet (SNMP) server, use port 161 or 162, depending on the configuration of the agent.

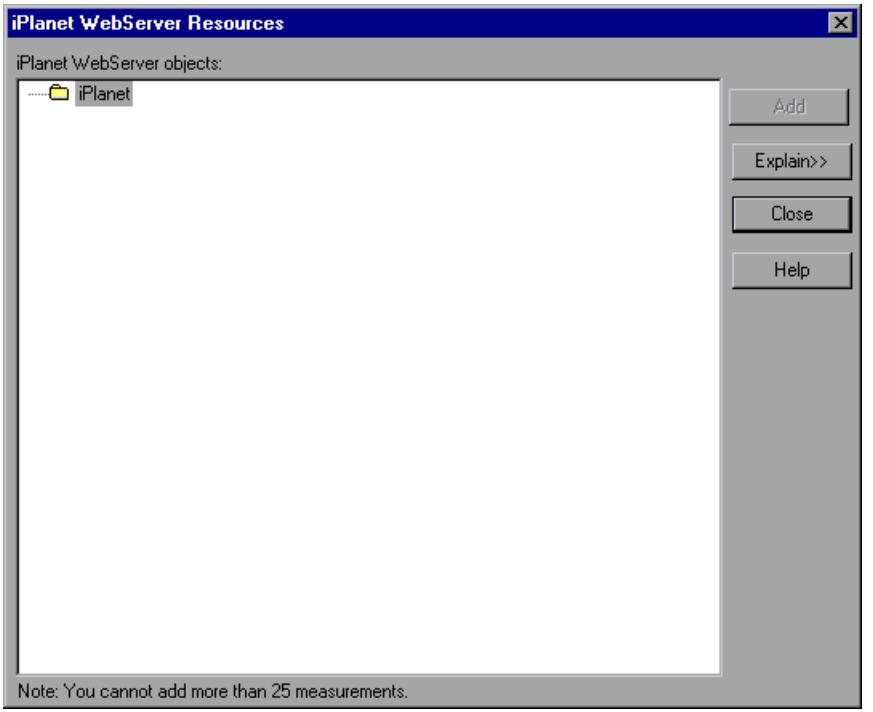
To configure the iPlanet (SNMP) Resources monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Web Server category, select **iPlanet (SNMP)**, and then click **Add**.

Note: You need to define the port number if the iPlanet SNMP agent is running on a different port than the default SNMP port. You can define the default port for your iPlanet server in the configuration file, *snmp.cfg*, located in *<LoadRunner Tuning Module root folder>\dat\monitors*. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the *snmp.cfg* file as follows:

```
; iPlanet (WebServer)
[cm_snmp_mon_iws60]
port=8888
```

The iPlanet WebServer Resources dialog box opens.



5 Browse the iPlanet WebServer Resources Object tree.

For a list of the available performance counters, see page 392.

6 To measure an object, select it, and click **Add**. For a description of each resource, click **Explain**>> to expand the dialog box. Add all the desired resources to the list, and click **Close**.

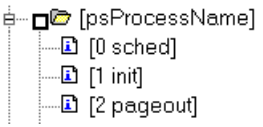
Note: The iPlanet (SNMP) monitor can only monitor up to 25 measurements.

---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: You can improve the level of measurement information for the iPlanet (SNMP) monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the *<LoadRunner Tuning Module root folder>\dat\monitors\snmp.cfg* file:
SNMP_show_string_nodes=1

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the iPlanet SNMP Add Measurements dialog box opens, the information is reread from the *snmp.cfg* file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

---

iPlanet (SNMP) Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
iwsInstanceTable	iPlanet Web Server instances
iwsInstanceEntry	iPlanet Web Server instances
iwsInstanceIndex	Server instance index
iwsInstanceId	Server instance identifier
iwsInstanceVersion	Server instance software version
iwsInstanceDescription	Description of server instance
iwsInstanceOrganization	Organization responsible for server instance
iwsInstanceContact	Contact information for person(s) responsible for server instance
iwsInstanceLocation	Location of server instance
iwsInstanceStatus	Server instance status
iwsInstanceUptime	Server instance uptime
iwsInstanceDeathCount	Number of times server instance processes have died
iwsInstanceRequests	Number of requests processed
iwsInstanceInOctets	Number of octets received
iwsInstanceOutOctets	Number of octets transmitted
iwsInstanceCount2xx	Number of 200-level (Successful) responses issued
iwsInstanceCount3xx	Number of 300-level (Redirection) responses issued
iwsInstanceCount4xx	Number of 400-level (Client Error) responses issued
iwsInstanceCount5xx	Number of 500-level (Server Error) responses issued

Measurement	Description
iwsInstanceCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued
iwsInstanceCount200	Number of 200 (OK) responses issued
iwsInstanceCount302	Number of 302 (Moved Temporarily) responses issued
iwsInstanceCount304	Number of 304 (Not Modified) responses issued
iwsInstanceCount400	Number of 400 (Bad Request) responses issued
iwsInstanceCount401	Number of 401 (Unauthorized) responses issued
iwsInstanceCount403	Number of 403 (Forbidden) responses issued
iwsInstanceCount404	Number of 404 (Not Found) responses issued
iwsInstanceCount503	Number of 503 (Unavailable) responses issued
iwsInstanceLoad 1MinuteAverage	System load average for 1 minute
iwsInstanceLoad 5MinuteAverage	System load average for 5 minutes
iwsInstanceLoad 15MinuteAverage	System load average for 15 minutes
iwsInstanceNetwork InOctets	Number of octets transmitted on the network per second
iwsInstanceNetwork OutOctets	Number of octets received on the network per second
iwsVsTable	iPlanet Web Server virtual servers
iwsVsEntry	iPlanet Web Server virtual server
iwsVsIndex	Virtual server index
iwsVsId	Virtual server identifier
iwsVsRequests	Number of requests processed
iwsVsInOctets	Number of octets received
iwsVsOutOctets	Number of octets transmitted

Measurement	Description
iwsVsCount2xx	Number of 200-level (Successful) responses issued
iwsVsCount3xx	Number of 300-level (Redirection) responses issued
iwsVsCount4xx	Number of 400-level (Client Error) responses issued
iwsVsCount5xx	Number of 500-level (Server Error) responses issued
iwsVsCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued
iwsVsCount200	Number of 200 (OK) responses issued
iwsVsCount302	Number of 302 (Moved Temporarily) responses issued
iwsVsCount304	Number of 304 (Not Modified) responses issued
iwsVsCount400	Number of 400 (Bad Request) responses issued
iwsVsCount401	Number of 401 (Unauthorized) responses issued
iwsVsCount403	Number of 403 (Forbidden) responses issued
iwsVsCount404	Number of 404 (Not Found) responses issued
iwsVsCount503	Number of 503 (Unavailable) responses issued
iwsProcessTable	iPlanet Web Server processes
iwsProcessEntry	iPlanet Web Server process
iwsProcessIndex	Process index
iwsProcessId	Operating system process identifier
iwsProcessThreadCount	Number of request processing threads
iwsProcessThreadIdle	Number of request processing threads currently idle
iwsProcessConnection QueueCount	Number of connections currently in connection queue

Measurement	Description
iwsProcessConnection QueuePeak	Largest number of connections that have been queued simultaneously
iwsProcessConnection QueueMax	Maximum number of connections allowed in connection queue
iwsProcessConnection QueueTotal	Number of connections that have been accepted
iwsProcessConnection QueueOverflows	Number of connections rejected due to connection queue overflow
iwsProcessKeepalive Count	Number of connections currently in keepalive queue
iwsProcessKeepaliveMax	Maximum number of connections allowed in keepalive queue
iwsProcessSizeVirtual	Process size in kbytes
iwsProcessSizeResident	Process resident size in kbytes
iwsProcessFraction SystemMemoryUsage	Fraction of process memory in system memory
iwsListenTable	iPlanet Web Server listen sockets
iwsListenEntry	iPlanet Web Server listen socket
iwsListenIndex	Listen socket index
iwsListenId	Listen socket identifier
iwsListenAddress	Address socket is listening on
iwsListenPort	Port socket is listening on
iwsListenSecurity	Encryption support
iwsThreadPoolTable	iPlanet Web Server thread pools
iwsThreadPoolEntry	iPlanet Web Server thread pool
iwsThreadPoolIndex	Thread pool index
iwsThreadPoolId	Thread pool identifier
iwsThreadPoolCount	Number of requests queued

Measurement	Description
iwsThreadPoolPeak	Largest number of requests that have been queued simultaneously
iwsThreadPoolMax	Maximum number of requests allowed in queue
iwsCpuTable	iPlanet Web Server CPUs
iwsCpuEntry	iPlanet Web Server CPU
iwsCpuIndex	CPU index
iwsCpuId	CPU identifier
iwsCpuIdleTime	CPU Idle Time
iwsCpuUserTime	CPU User Time
iwsCpuKernelTime	CPU Kernel Time

Monitoring Using a Proxy Server

LoadRunner Tuning Module allows you to monitor using the Apache and Netscape monitors when there is a proxy server between the Console and the monitored server. To enable this, you must define settings in your configuration file: in *<LoadRunner Tuning Module root folder>\dat\monitors\apache.cfg* for the Apache monitor, or in *<LoadRunner Tuning Module root folder>\dat\monitors\Netscape.cfg* for the Netscape monitor.

Before defining settings, you need to determine whether you want LoadRunner Tuning Module to obtain proxy settings from your Internet Explorer connection configuration, or from the proxy settings in the configuration file.

To have LoadRunner Tuning Module read proxy settings from your Internet Explorer connection:

1. In the Proxy Settings section of the configuration file, assign **useProxy** a value of 1.
2. If the proxy requires a username, password, or domain, enter these parameters on the lines **proxyUsername**, **proxyPassword**, and **proxyDomain**.

To have LoadRunner Tuning Module read proxy settings from the configuration file:

1 In the Proxy Settings section of the configuration file, enter the proxy information on the **httpProxy** line. Use the format: [<protocol>=][<scheme>://]<proxy>[:<port>][[<protocol>=][<scheme>://]<proxy>[:<port>]]

For example:
httpProxy=http=http://my_http_proxy:8080 https=https://my_https_proxy:9000

2 If the proxy requires a username, password, or domain, enter these parameters on the lines **proxyUsername**, **proxyPassword**, and **proxyDomain**.

To have LoadRunner Tuning Module connect directly to the server (any proxy settings are ignored):

In the Proxy Settings section of the configuration file, assign **useProxy** a value of 0.

33

Web Application Server Resource Monitoring

You can monitor a Web application server during a session step run and isolate application server performance bottlenecks using LoadRunner Tuning Module's Web Application Server Resource monitors.

This chapter describes:

- Configuring the Ariba Monitor
- Configuring the ATG Dynamo Monitor
- Configuring the BroadVision Monitor
- Configuring the ColdFusion Monitor
- Configuring the Fujitsu INTERSTAGE Monitor
- Configuring the iPlanet (NAS) Monitor
- Configuring the Microsoft Active Server Pages Monitor
- Configuring the Oracle9iAS HTTP Monitor
- Configuring the SilverStream Monitor
- Configuring the WebLogic (SNMP) Monitor
- Configuring the WebLogic (JMX) Monitor
- Configuring the WebSphere Monitor
- Configuring the WebSphere (EPM) Monitor

About Web Application Server Resource Monitors

Web Application Server Resource monitors provide you with information about the resource usage of the Ariba, ATG Dynamo, BroadVision, ColdFusion, Fujitsu INTERSTAGE, iPlanet (NAS), Microsoft ASP, Oracle9iAS HTTP, SilverStream, WebLogic (SNMP), WebLogic (JMX), and WebSphere application servers during session step execution. In order to obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the session step.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Configuring the Ariba Monitor

You select measurements to monitor the Ariba server using the Ariba Monitor Configuration dialog box.

Note: The port you use to monitor an Ariba server through a firewall depends on the configuration of your server.

To configure the Ariba monitor:

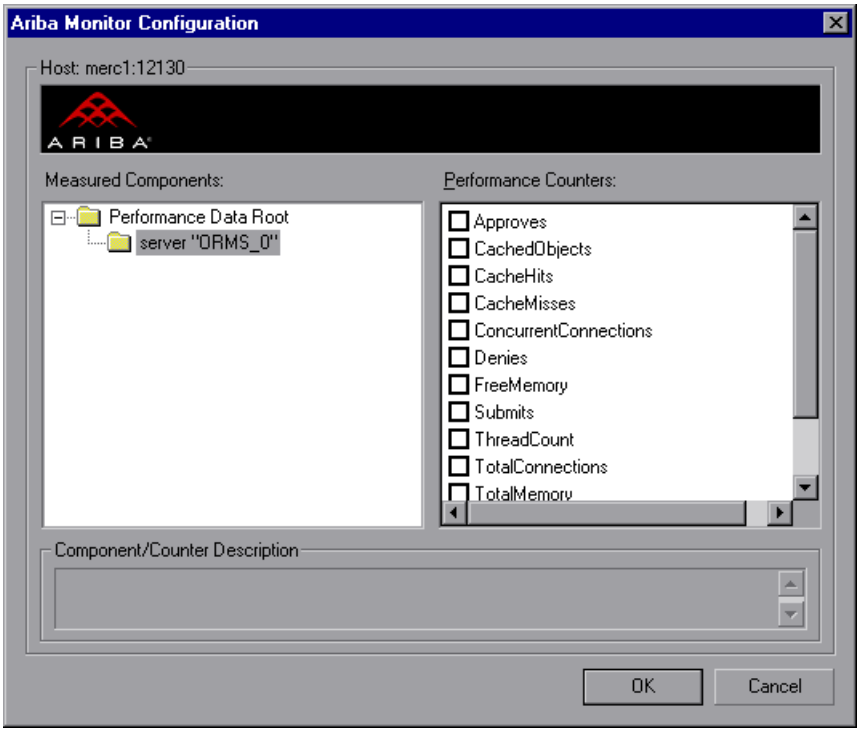
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **Ariba**, and then click **Add**.

The Ariba Monitor Configuration dialog box opens.



5 Browse the Measured Components tree. Check the required performance counters in the Ariba Monitor Configuration window's right pane.

For a list of the available performance counters, see page 403.

6 Click **OK** in the Ariba Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the Ariba monitor.

XML Accessibility Verification

Only browsers that are XML-compatible will allow you to view the performance XML file.

To verify whether the XML file is accessible:

Display the XML file through the browser. The URL should be in the following format: http://<*server name*:*port number*>/metrics?query=getStats

For example: http://merc1:12130/metrics?query=getStats

Note: In some cases, although the browser is XML-compatible, it may still return the error: The XML page cannot be displayed. In these cases, the XML file can be accessed by the Ariba performance monitor, although it cannot be viewed by the browser.

Ariba Performance Counters

The following tables describe the counters that can be monitored:

Core Server Performance Counters

Measurement	Description
Requisitions Finished	The instantaneous reading of the length of the worker queue at the moment this metric is obtained. The longer the worker queue, the more user requests are delayed for processing.
Worker Queue Length	The instantaneous reading of the length of the worker queue at the moment this metric is obtained. The longer the worker queue, the more user requests are delayed for processing.
Concurrent Connections	The instantaneous reading of the number of concurrent user connections at the moment this metric is obtained
Total Connections	The cumulative number of concurrent user connections since Ariba Buyer was started.
Total Memory	The instantaneous reading of the memory (in KB) being used by Ariba Buyer at the moment this metric is obtained
Free Memory	The instantaneous reading of the reserved memory (in bytes) that is not currently in use at the moment this metric is obtained
Up Time	The amount of time (in hours and minutes) that Ariba Buyer has been running since the previous time it was started
Number of Threads	The instantaneous reading of the number of server threads in existence at the moment this metric is obtained
Number of Cached Objects	The instantaneous reading of the number of Ariba Buyer objects being held in memory at the moment this metric is obtained

Measurement	Description
Average Session Length	The average length of the user sessions (in seconds) of all users who logged out since previous sampling time. This value indicates on average how long a user stays connected to server.
Average Idle Time	The average idle time (in seconds) for all the users who are active since previous sampling time. The idle time is the period of time between two consecutive user requests from the same user.
Approves	The cumulative count of the number of approves that happened during the sampling period. An Approve consists of a user approving one Approvable.
Submits	The cumulative count of the number of Approvables submitted since previous sampling time
Denies	The cumulative count of the number of submitted Approvables denied since previous sampling time
Object Cache Accesses	The cumulative count of accesses (both reads and writes) to the object cache since previous sampling time
Object Cache Hits	The cumulative count of accesses to the object cache that are successful (cache hits) since previous sampling time

System Related Performance Counters

Measurement	Description
Database Response Time	The average response time (in seconds) to the database requests since the previous sampling time
Buyer to DB server Traffic	The cumulative number of bytes that Ariba Buyer sent to DB server since the previous sampling time.
DB to Buyer server Traffic	The cumulative number of bytes that DB server sent to Ariba Buyer since the previous sampling time

Measurement	Description
Database Query Packets	The average number of packets that Ariba Buyer sent to DB server since the previous sampling time
Database Response Packets	The average number of packets that DB server sent to Ariba Buyer since the previous sampling time

Configuring the ATG Dynamo Monitor

The ATG Dynamo monitor uses SNMP to retrieve ATG Dynamo server statistics. You define the measurements for the ATG Dynamo monitor using the ATG Dynamo Resources dialog box.

To configure the ATG Dynamo server monitor:

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

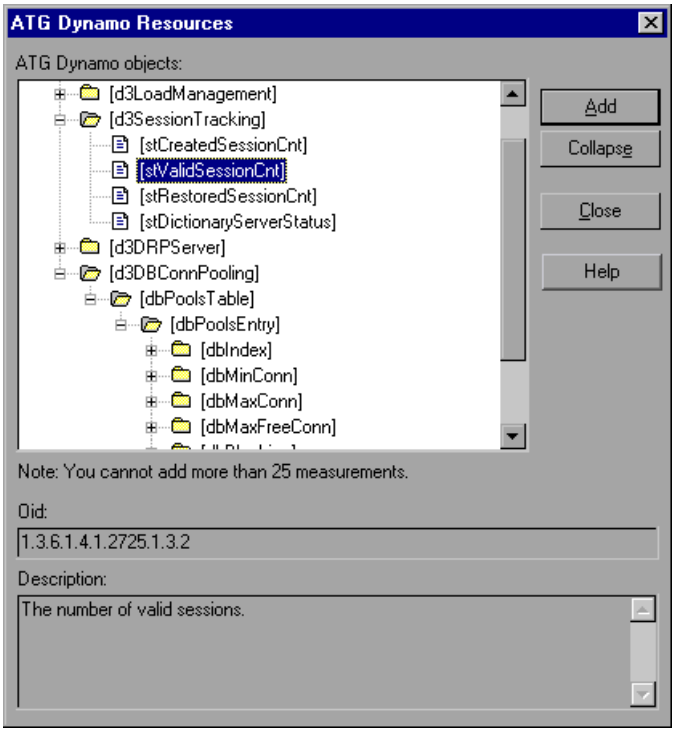
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **ATG Dynamo**, and then click **Add**.

Note: You need to define the port number if the ATG SNMP agent is running on a different port than the default ATG SNMP port 8870. You can define the default port for your ATG server in the configuration file, *snmp.cfg*, located in *<LoadRunner Tuning Module root folder>\dat\monitors*. For example, if the port used by the SNMP agent on your ATG system is 8888, you should edit the *snmp.cfg* file as follows:

```
; ATG Dynamo
[cm_snmp_mon_atg]
port=8888
```

The ATG Dynamo Resources dialog box opens.



5 Browse the ATG Dynamo Object tree, and select the measurements you want to monitor. For a list of the available performance counters, see page 407.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

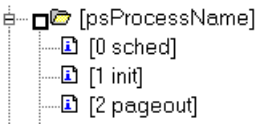
Note: The ATG Dynamo monitor can only monitor up to 25 measurements.

---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: You can improve the level of measurement information for the ATG Dynamo monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the *<LoadRunner Tuning Module root folder>\dat\monitors\snmp.cfg* file:
SNMP_show_string_nodes=1

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the ATG Dynamo Add Measurements dialog box opens, the information is reread from the *snmp.cfg* file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

ATG Dynamo Performance Counters

The following tables describe the measurements that can be monitored:

d3System

Measurement	Description
sysTotalMem	The total amount of memory currently available for allocating objects, measured in bytes
sysFreeMem	An approximation of the total amount of memory currently available for future allocated objects, measured in bytes
sysNumInfoMsgs	The number of system global info messages written
sysNumWarningMsgs	The number of system global warning messages written
sysNumErrorMsgs	The number of system global error messages written

d3LoadManagement

Measurement	Description
lmIsManager	True if the Dynamo is running a load manager
lmManagerIndex	Returns the Dynamo's offset into the list of load managing entities
lmIsPrimaryManager	True if the load manager is an acting primary manager
lmServicingCMs	True if the load manager has serviced any connection module requests in the amount of time set as the connection module polling interval
lmCMLDRPPort	The port of the connection module agent
lmIndex	A unique value for each managed entity
lmSNMPPort	The port for the entry's SNMP agent
lmProbability	The probability that the entry will be given a new session

Measurement	Description
lmNewSessions	Indicates whether or not the entry is accepting new sessions, or if the load manager is allowing new sessions to be sent to the entry. This value is inclusive of any override indicated by lmNewSessionOverride.
lmNewSessionOverride	The override set for whether or not a server is accepting new sessions

d3SessionTracking

Measurement	Description
stCreatedSessionCnt	The number of created sessions
stValidSessionCnt	The number of valid sessions
stRestoredSessionCnt	The number of sessions migrated to the server
StDictionaryServerStatus	d3Session Tracking

d3DRPServer

Measurement	Description
drpPort	The port of the DRP server
drpTotalReqsServed	Total number of DRP requests serviced
drpTotalReqTime	Total service time in msecs for all DRP requests
drpAvgReqTime	Average service time in msecs for each DRP request
drpNewessions	True if the Dynamo is accepting new sessions

d3DBConnPooling

Measurement	Description
dbPoolsEntry	A pooling service entry containing information about the pool configuration and current status
dbIndex	A unique value for each pooling service

Measurement	Description
dbPoolID	The name of the DB connection pool service
dbMinConn	The minimum number of connections pooled
dbMaxConn	The maximum number of connections pooled
dbMaxFreeConn	The maximum number of free pooled connections at a time
dbBlocking	Indicates whether or not the pool is to block out check outs
dbConnOut	Returns the number of connections checked out
dbFreeResources	Returns the number of free connections in the pool. This number refers to connections actually created that are not currently checked out. It does not include how many more connections are allowed to be created as set by the maximum number of connections allowed in the pool.
dbTotalResources	Returns the number of total connections in the pool. This number refers to connections actually created and is not an indication of how many more connections may be created and used in the pool.

Configuring the BroadVision Monitor

To monitor a BroadVision server, you must grant the client permission to invoke or launch services on the server.

Note: The port you use to monitor a BroadVision server through a firewall depends on the configuration of your server.

To grant permission for a BroadVision server:

➤ Use the Iona Technologies (Orbix) command for setting user and access permission on a load generator machine:

chmodit [-h <host>] [-v] { <server> | -a <dir> }

{i{+,-}{user,group} | l{+,-}{user,group} }

➤ If you experience problems connecting to the BroadVision monitor, you may need to redefine the permissions to "all."

To invoke permission for all, enter the following command at the BroadVision server command prompt:

chmodit <server> i+all

To launch permission for all, enter the following command at the BroadVision server command prompt:

chmodit <server> l+all

➤ Alternatively, set ORBIX_ACL. Setting ORBIX_ACL=i+all l+all in the BroadVision/Orbix configuration file gives permission to all.

In addition, to monitor a BroadVision server, you need to have JDK 1.2 or higher installed on the Console machine.

You can install JDK 1.2 by following the download and installation instructions at the following Web site: http://java.sun.com/products/jdk/1.2/

Before activating the monitor, make sure that your Java environment is configured properly.

To configure your Java environment:

1. Open the Windows Registry.
2. The registry should contain the correct path to the Java executable (java.exe) under the JDK 1.2 installation directory. Verify the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\java.exe

3. The registry should contain the correct path to the Java run-time environment (JRE) under the JRE 1.2 installation directory. Verify the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment\1.2\JavaHome

To configure the BroadVision online monitor:

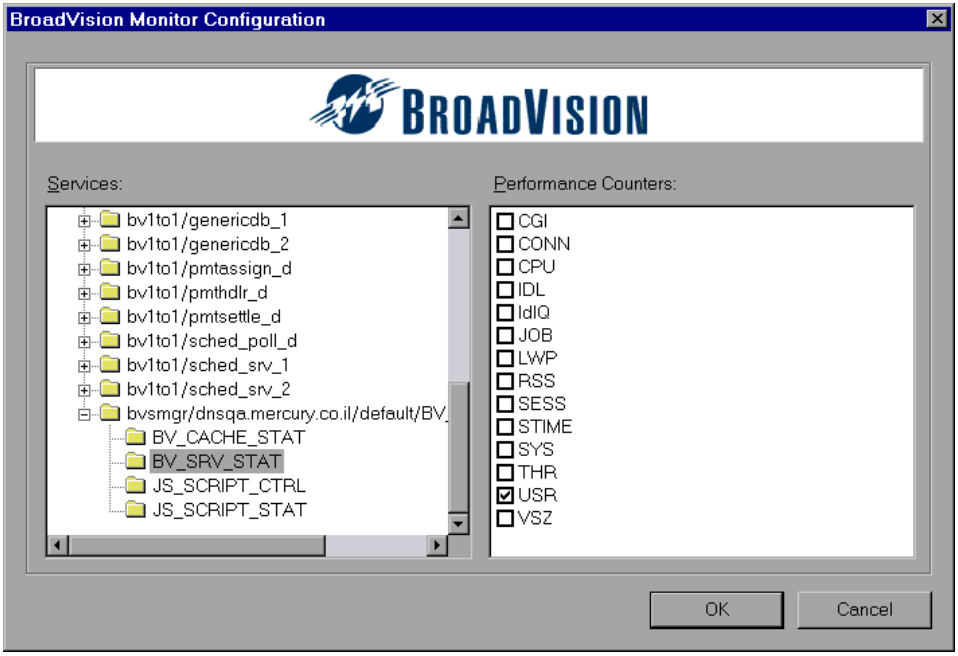
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **BroadVision (5.5 - 6.x)** or **BroadVision (4.5 -5.4)**, and then click **Add**.

The BroadVision Monitor Configuration dialog box opens, displaying the available measurements:



5 Browse the Services tree and check the required performance counters in the BroadVision Monitor Configuration window's right pane.

For a list of the available performance counters, see page 413.

6 Click **OK** in the BroadVision Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the BroadVision monitor.

BroadVision Performance Counters

The following table describes the servers/services that can be monitored:

Server	Multiple Instances	Description
adm_srv	No	One-To-One user administration server. There must be one.
alert_srv	No	Alert server handles direct IDL function calls to the Alert system.
bvconf_srv	No	One-To-One configuration management server. There must be one.
cmsdb	Yes	Visitor management database server.
cntdb	Yes	Content database server.
deliv_smtp_d	Yes	Notification delivery server for e-mail type messages. Each instance of this server must have its own ID, numbered sequentially starting with "1".
deliv_comp_d	No	Notification delivery completion processor.
extdbacc	Yes	External database accessor. You need at least one for each external data source.
genericdb	No	Generic database accessor handles content query requests from applications, when specifically called from the application. This is also used by the One-To-One Command Center.
hostmgr	Yes	Defines a host manager process for each machine that participates in One-To-One, but doesn't run any One-To-One servers. For example, you need a hostmgr on a machine that runs only servers. You don't need a separate hostmgr on a machine that already has one of the servers in this list.

Server	Multiple Instances	Description
g1_ofbe_srv	No	Order fulfillment back-end server.
g1_ofdb	Yes	Order fulfillment database server.
g1_om_srv	No	Order management server.
pmtassign_d	No	The payment archiving daemon routes payment records to the archives by periodically checking the invoices table, looking for records with completed payment transactions, and then moving those records into an archive table.
pmthdlr_d	Yes	For each payment processing method, you need one or more authorization daemons to periodically acquire the authorization when a request is made.
pmtsettle_d	Yes	Payment settlement daemon periodically checks the database for orders of the associated payment processing method that need to be settled, and then authorizes the transactions.
sched_poll_d	No	Notification schedule poller scans the database tables to determine when a notification must be run.
sched_srv	Yes	Notification schedule server runs the scripts that generate the visitor notification messages.

Performance Counters

Performance counters for each server/service are divided into logical groups according to the service type.

The following section describes all the available counters under each group. Note that the same group can have a different number of counters, depending on the service.

Counter groups:

- BV_DB_STAT
- BV_SRV_CTRL

- BV_SRV_STAT
- NS_STAT
- BV_CACHE_STAT
- JS_SCRIPT_CTRL
- JS_SCRIPT_STAT

BV_DB_STAT

The database accessor processes have additional statistics available from the BV_DB_STAT memory block. These statistics provide information about database accesses, including the count of selects, updates, inserts, deletes, and stored procedure executions.

- DELETE - Count of deletes executions
- INSERT - Count of inserts executions
- SELECT - Count of selects executions
- SPROC - Count of stored procedure executions.
- UPDATE - Count of updates executions

BV_SRV_CTRL

- SHUTDOWN

NS_STAT

The NS process displays the namespace for the current One-To-One environment, and optionally can update objects in a name space.

- Bind
- List
- New
- Rebnd
- Rsolv
- Unbnd

BV_SRV_STAT

The display for Interaction Manager processes includes information about the current count of sessions, connections, idle sessions, threads in use, and count of CGI requests processed.

- **HOST** - Host machine running the process.
- **ID** - Instance of the process (of which multiple can be configured in the *bv1to1.conf* file), or engine ID of the Interaction Manager.
- **CGI** - Current count of CGI requests processed.
- **CONN** - Current count of connections.
- **CPU** - CPU percentage consumed by this process. If a process is using most of the CPU time, consider moving it to another host, or creating an additional process, possibly running on another machine. Both of these specifications are done in the *bv1to1.conf* file. The CPU % reported is against a single processor. If a server is taking up a whole CPU on a 4 processor machine, this statistic will report 100%, while the Windows Task Manager will report 25%. The value reported by this statistic is consistent with "% Processor Time" on the Windows Performance Monitor.
- **GROUP** - Process group (which is defined in the *bv1to1.conf* file), or Interaction Manager application name.

- **STIME** - Start time of server. The start times should be relatively close. Later times might be an indication that a server crashed and was automatically restarted.
- **IDL** - Total count of IDL requests received, not including those to the monitor.
- **IdlQ**
- **JOB**
- **LWP** - Number of light-weight processes (threads).
- **RSS** - Resident memory size of server process (in kilobytes).
- **STIME** - System start time.
- **SESS** - Current count of sessions.
- **SYS** - Accumulated system mode CPU time (seconds).
- **THR** - Current count of threads.
- **USR** - Accumulated user mode CPU time (seconds).
- **VSZ** - Virtual memory size of server process (in kilobytes). If a process is growing in size, it probably has a memory leak. If it is an Interaction Manager process, the culprit is most likely a component or dynamic object (though Interaction Manager servers do grow and shrink from garbage collection during normal use).

BV_CACHE_STAT

Monitors the request cache status.

The available counters for each request are:

- **CNT- Request_Name-HIT** - Count of requests found in the cache.
- **CNT- Request_Name-MAX** - Maximum size of the cache in bytes
- **CNT- Request_Name-SWAP** - Count of items that got swapped out of the cache.
- **CNT- Request_Name-MISS** - Count of requests that were not in the cache.
- **CNT- Request_Name-SIZE** - Count of items currently in the cache.

Cache Metrics

Cache metrics are available for the following items:

- **AD**
- **ALERTSCHED** - Notification schedules are defined in the BV_ALERTSCHED and BV_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
- **CATEGORY_CONTENT**
- **DISCUSSION** - The One-To-One discussion groups provide moderated system of messages and threads of messages aligned to a particular topic. Use the Discussion group interfaces for creating, retrieving and deleting individual messages in a discussion group. To create, delete, or retrieve discussion groups, use the generic content management API. The BV_DiscussionDB object provides access to the threads and messages in the discussion group database.
- **EXT_FIN_PRODUCT**
- **EDITORIAL** - Using the Editorials content module, you can point cast and community cast personalized editorial content, and sell published text on your One-To-One site. You can solicit editorial content, such as investment reports and weekly columns, from outside authors and publishers, and create your own articles, reviews, reports, and other informative media. In addition to text, you can use images, sounds, music, and video presentations as editorial content.
- **INCENTIVE** - Contains sales incentives
- **MSGSCHED** - Contains the specifications of visitor-message jobs. Notification schedules are defined in the BV_ALERTSCHED and BV_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
- **MSGSCRIPT** - Contains the descriptions of the JavaScripts that generate visitor messages and alert messages. Contains the descriptions of the JavaScripts that generate targeted messages and alert messages. Use the Command Center to add message script information to this table by selecting the Visitor Messages module in the Notifications group. For more information, see the Command Center User's Guide.

- **PRODUCT** - BV_PRODUCT contains information about the products that a visitor can purchase.
- **QUERY** - BV_QUERY contains queries.
- **SCRIPT** - BV_SCRIPT contains page scripts.
- **SECURITIES**
- **TEMPLATE** - The Templates content module enables you to store in the content database any BroadVision page templates used on your One-To-One site. Combining BroadVision page templates with BroadVision dynamic objects in the One-To-One Design Center application is one way for site developers to create One-To-One Web sites. If your developers use these page templates, you can use the Command Center to enter and manage them in your content database. If your site doesn't use BroadVision page template, you will not use this content module.

JS_SCRIPT_CTRL

- CACHE
- DUMP
- FLUSH
- METER
- TRACE

JS_SCRIPT_STAT

- ALLOC
- ERROR
- FAIL
- JSPPERR
- RELEASE
- STOP
- SUCC
- SYNTAX

Configuring the ColdFusion Monitor

You select measurements to monitor the ColdFusion server using the ColdFusion dialog box.

Note: The ColdFusion monitor works via HTTP and supports UNIX platforms. If you want to monitor the ColdFusion server on Windows platforms, you can also use the Windows Resource monitor.

To set up the ColdFusion monitor environment:

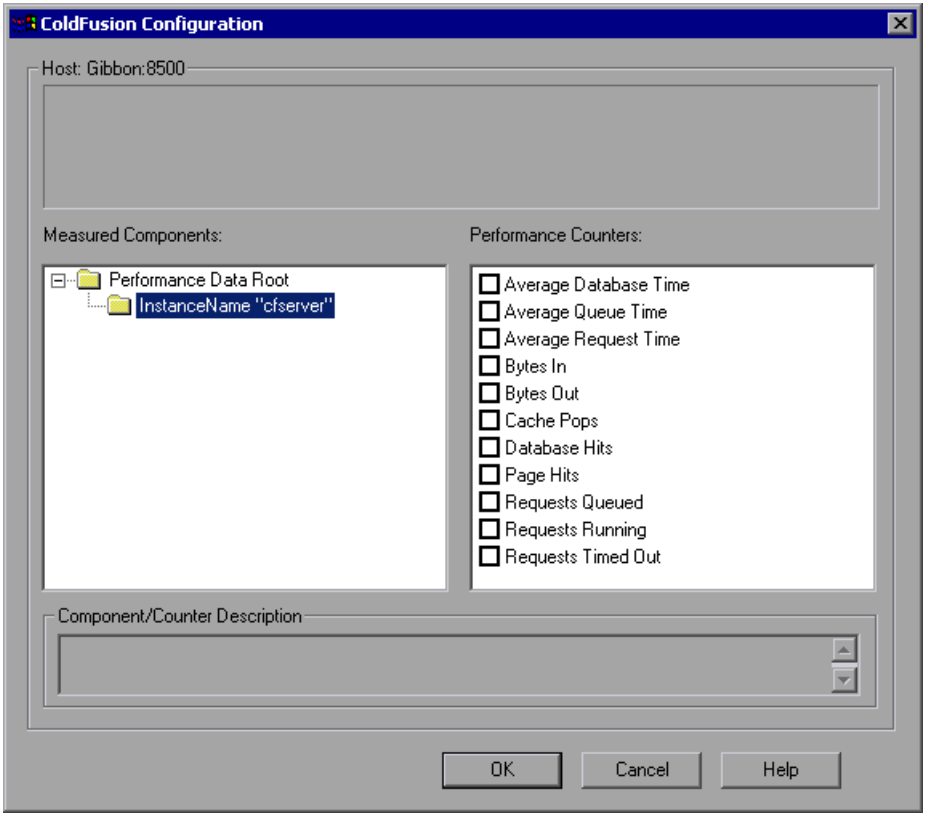
Copy the *<LoadRunner Tuning Module installation>\dat\monitors\perfmon.cfm* file into the *<ColdFusion Home>\cfide\administrator* directory. By default, the ColdFusion monitor checks for the *<ColdFusion Home>\cfide\administrator\perfmon.cfm* file.

Note: The port you use to monitor a ColdFusion server through a firewall depends on the configuration of your server.

To configure the ColdFusion monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **ColdFusion**, and then click **Add**.
5. If you are prompted for a username and password, enter them and click **OK**.

The ColdFusion Configuration dialog box displays the available measurements.



6 Browse the Measured Components tree, and check the required performance counters in the ColdFusion Monitor Configuration window's right pane.

For a list of the available performance counters, see page 422.

7 Click **OK** in the ColdFusion Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the ColdFusion monitor.

ColdFusion Performance Counters

The following table describes the default counters that can be measured:

Measurement	Description
Avg. Database Time (msec)	The running average of the amount of time, in milliseconds, that it takes ColdFusion to process database requests.
Avg. Queue Time (msec)	The running average of the amount of time, in milliseconds, that requests spent waiting in the ColdFusion input queue before ColdFusion began to process the request.
Avg Req Time (msec)	The running average of the total amount of time, in milliseconds, that it takes ColdFusion to process a request. In addition to general page processing time, this value includes both queue time and database processing time.
Bytes In/sec	The number of bytes per second sent to the ColdFusion server.
Bytes Out/sec	The number of bytes per second returned by the ColdFusion server.
Cache Pops	Cache pops.
Database Hits/sec	This is the number of database hits generated per second by the ColdFusion server.
Page Hits/sec	This is the number of Web pages processed per second by the ColdFusion server.
Queued Requests	The number of requests currently waiting to be processed by the ColdFusion server.
Running Requests	The number of requests currently being actively processed by the ColdFusion server.
Timed Out Requests	The number of requests that timed out due to inactivity timeouts.

Configuring the Fujitsu INTERSTAGE Monitor

The Fujitsu INTERSTAGE monitor uses SNMP to retrieve Fujitsu INTERSTAGE server statistics. You define the measurements for the Fujitsu INTERSTAGE monitor using the Fujitsu INTERSTAGE SNMP Resources dialog box.

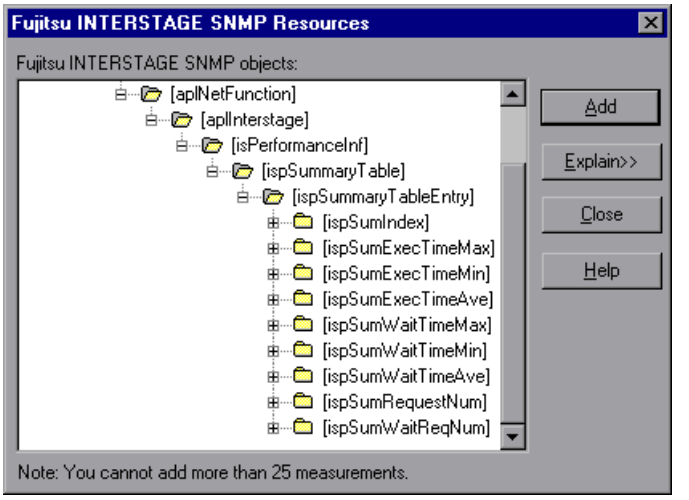
To configure the Fujitsu INTERSTAGE server monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **Fujitsu INTERSTAGE**, and then click **Add**.

Note: You need to define the port number if the Fujitsu INTERSTAGE SNMP agent is running on a different port than the default SNMP port 161. You can define the default port for your Fujitsu INTERSTAGE server in the configuration file, *snmp.cfg*, located in *<LoadRunner Tuning Module root folder>\dat\monitors*. For example, if the port used by the SNMP agent on your Fujitsu INTERSTAGE system is 8888, you should edit the *snmp.cfg* file as follows:

```
; Fujitsu INTERSTAGE
[cm_snmp_mon_isp]
port=8888
```

The Fujitsu INTERSTAGE SNMP Resources dialog box opens.



5 Browse the Fujitsu INTERSTAGE SNMP Object tree, and select the measurements you want to monitor.

For a list of the available performance counters, see page 425.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

Note: The Fujitsu INTERSTAGE monitor can only monitor up to 25 measurements.

---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Fujitsu INTERSTAGE Performance Counters

The following tables describe the measurements that can be monitored:

Measurement	Description
IspSumObjectName	The object name of the application for which performance information is measured
IspSumExecTimeMax	The maximum processing time of the application within a certain period of time
IspSumExecTimeMin	The minimum processing time of the application within a certain period of time
IspSumExecTimeAve	The average processing time of the application within a certain period of time
IspSumWaitTimeMax	The maximum time required for INTERSTAGE to start an application after a start request is issued
IspSumWaitTimeMin	The minimum time required for INTERSTAGE to start an application after a start request is issued
IspSumWaitTimeAve	The average time required for INTERSTAGE to start an application after a start request is issued
IspSumRequestNum	The number of requests to start an application
IspSumWaitReqNum	The number of requests awaiting application activation

Configuring the iPlanet (NAS) Monitor

The iPlanet (NAS) monitor uses the SNMP to retrieve iPlanet (NAS) server statistics. You define the measurements for the iPlanet (NAS) monitor using the iPlanet (NAS) dialog box.

Setting up the iPlanet (NAS) Application Server

This section offers a short explanation on setting up SNMP monitoring of the iPlanet Application Server. It is intended to supplement the iPlanet documentation, not act as a replacement. For an explanation of the SNMP reporting architecture and theory, refer to the iPlanet documentation.

Note: The instructions below assume that SNMP statistics will be collected on the standard SNMP port 161.

SNMP Summary

- Solaris has a native SNMP agent, "snmpdx", that is started automatically at boot time by the script /etc/rc3.d/S76snmpdx. This daemon communicates on the standard SNMP port 161. The port number can be changed with the -p <port> option.
- Planet Products are shipped with their own SNMP agents. The architecture is such that there is one "master agent" per host, which a network management station communicates with, and one or more "subagents" that collect data from various iPlanet products and forward statistics to the master agent. The master agent also defaults to communicating on port 161.
- To run both the Solaris SNMP agent and the iPlanet SNMP agent, a proxy must be used that makes the Sun agent look like a subagent to the iPlanet master agent.

Steps Overview

- Login to the system as root.
- Change the port number for the Solaris SNMP agent.
- Configure and run the iPlanet agents "magt" and "sagt".
- Start the Solaris SNMP agent.
- Configure iPlanet Application Server for SNMP statistics.
- Start SNMP subagents for iPlanet Directory Server and iPlanet Web Server (optional).

To change the port number for the Solaris SNMP agent:

1. Login to the system as root. (Only a root user can change the port number and run the agents).
2. Stop the SNMP agent by running /etc/rc2.d/K76snmpdx stop.

3 Edit /etc/rc3.d/S76snmpdx to run the Solaris daemon on a non-standard port number. For example, 1161:
Replace
/usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
with
/usr/lib/snmp/snmpdx -p 1161 -y -c /etc/snmp/conf

To configure and run the iPlanet agents "magt" and "sagt":

The master and proxy agents and startup scripts are found in *<ias install directory>*/snmp.

1 In the script S75snmpagt, add a line to the environment variable GX_ROOTDIR so that it points to your iAS installation. For example, if the iPlanet Application Server is installed in /usr/iplanet/ias6/ias:

```
GX_ROOTDIR=/usr/iplanet/ias6/ias
exprt GX_ROOTDIR
```

2 Copy the script S75snmpagt to */etc/rc3.d*

3 chmod 755 /etc/rc3.d/S75snmpagt

4 In /etc/rc3.d/S75snmpagt /etc/rc2.d/K07snmpagt

5 You can configure system information and traps.

In the example below, information has been added about the system owner and location, and SNMP traps have been sent to a network manager station ("mde.uk.sun.com").

```
COMMUNITY       public
ALLOW ALL OPERATIONS
INITIAL sysLocation "Under Joe Bloggs' Desk in Headquarters"
INITIAL sysContact "Joe Bloggs
Email: Joe.Bloggs@Sun.COM
Voice: +1 650 555 1212"
MANAGER mde.uk.sun.com
SEND ALL TRAPS TO PORT 162
WITH COMMUNITY public
```

Note: There is no need to edit the proxy agent's configuration file (CONFIG_SAGT).

---

6 Start the iPlanet agents by running the command:
/etc/rc3.d/S75snmpagt start

To start the Solaris SNMP agent:

Restart the Solaris SNMP agent by running the command:
/etc/rc3.d/S76snmpdx start

To configure iPlanet Application Server for SNMP statistics:

1 Start the iPlanet Application Server admin tool ksvradmin.

2 In the General View, select the instance name that you want to manage.

3 Click the **SNMP** tab in the management frame.

4 Select **Enable SNMP Administration and Monitoring** and **Enable SNMP Debug**.

5 Type "60" in the Connection Attempt Interval field, and exit ksvradmin.

6 Restart the iPlanet Application Server with the commands:

```
iascontrol stop
iascontrol kill
iascontrol start
```

7 Check in the log file <*ias install directory*>/logs/ias.log that the application server successfully connected to the master agent. You should see the following line:
kas> SNMP: Connected to master agent

To start SNMP subagents for iPlanet Web Server:

1 Use your Web browser to access the iPlanet Web Server.

2 Choose the Web server you wish to administer, and click the **Manage** button.

3 Select the **Monitor** tab, and click **SNMP Subagent Configuration** on the left side of the page.

4 Type in the configuration information and set the radio button **Enable SNMP Statistics Collection** to "On".

5 Click **SNMP Subagent Control**.

6 Click the **Start** button.

To start SNMP subagents for iPlanet Directory Server:

1 Use the Netscape Administration Console to manage the iPlanet Directory Server.

2 Select the **Configuration** tab.

3 Click the **SNMP** tab in the Configuration frame.

4 Select the **Enable statistics collection** check box.

5 Set "Master Host" to "localhost".

6 Set "Master port" to 199.

7 In the other fields, enter the appropriate information.

8 Click the **Start Subagent** button.

Summary note

Use your SNMP management tool to query the SNMP master agent on port 161. You should see all the information provided by the Solaris SNMP agent, as well as any iPlanet subagents that you have configured.

The next time that you boot Solaris, the Sun and iPlanet SNMP agents will be started automatically by the boot scripts which you have configured.

Configuring the iPlanet (NAS) Monitor in the Console

Once you have configured the iPlanet SNMP Service, you must select the counters that you want the iPlanet (NAS) monitor to measure. You select these measurements using the iPlanet (NAS) Resources dialog box.

To configure the iPlanet (NAS) Resources monitor:

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

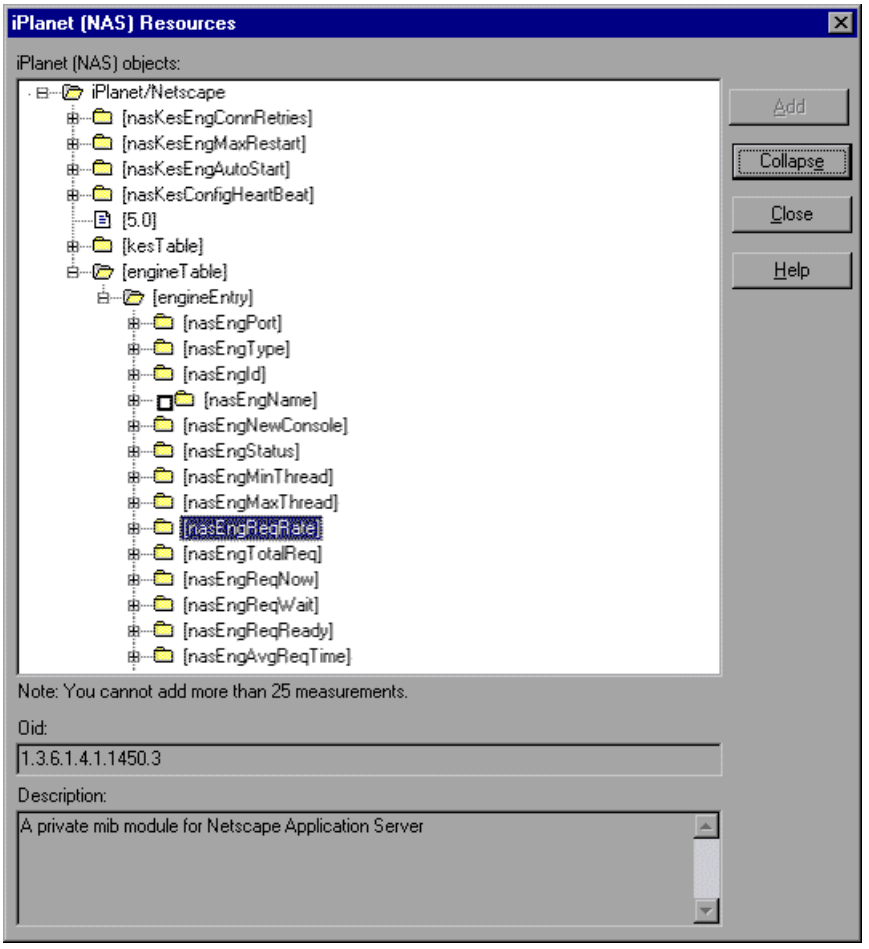
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **iPlanet (NAS)**, and then click **Add**.

Note: You need to define the port number if the iPlanet SNMP agent is running on a different port than the default SNMP port. You can define the default port for your iPlanet server in the configuration file, *snmp.cfg*, located in *<LoadRunner Tuning Module root folder>\dat\monitors*. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the *snmp.cfg* file as follows:

```
; iPlanet (NAS)
[cm_snmp_mon_nas]
port=8888
```

The iPlanet (NAS) Resources dialog box opens.



5 Browse the iPlanet (NAS) Resources Object tree.

For a list of the available performance counters, see page 432.

6 To measure an object, select it, and click **Add**. Add all the desired resources to the list, and click **Close**.

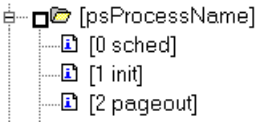
Note: The iPlanet (NAS) monitor can only monitor up to 25 measurements.

---

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: You can improve the level of measurement information for the iPlanet (NAS) monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the *<LoadRunner Tuning Module root folder>\dat\monitors\snmp.cfg* file:
SNMP_show_string_nodes=1

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the iPlanet (NAS) Add Measurements dialog box opens, the information is reread from the *snmp.cfg* file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

---

iPlanet (NAS) Performance Counters

The following tables describe the counters that can be monitored:

Netscape Performance Counters

Measurement	Description
nasKesEngConnRetries	The maximum number of times the administration server will try to connect to an engine.
nasKesEngMaxRestart	The maximum number of times the administration server will restart an engine after a failure.
nasKesEngAutoStart	Start all the engines at startup of the administration server.
nasKesConfigHeartBeat	Heart Beat.

KES Performance Counters

Measurement	Description
nasKesId	The ID of the KES this engine belongs to.
nasKesMinThread	The default minimum number of threads per engine.
nasKesMaxThread	The default maximum number of threads per engine.
nasKesLoadBalancerDisable	Enable or Disable the load balancer service.
nasKesCpuLoad	The total CPU usage on this host.
nasKesDiskLoad	The total disk usage on this host.
nasKesMemLoad	The total memory usage on this host.
nasKesRequestLoad	The number of requests on this NAS.
nasKesCpuLoadFactor	The relative importance of CPU usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.
nasKesDiskLoadFactor	The relative importance of Disk usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.

Measurement	Description
nasKesMemLoadFactor	The relative importance of Memory usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.
nasKesAppLogicsRunningFactor	The relative importance of the number of times an AppLogic is run in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesResultsCachedFactor	The relative importance of the cached results of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%
nasKesAvgExecTimeFactor	The relative importance of the average execution time of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesLastExecTimeFactor	The relative importance of the last execution time of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesHitsFactor	The relative importance of the number of AppLogics running in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.

Measurement	Description
nasKesServerLoadFactor	The relative importance of the server load (computed using the four server load factors) in computing AppLogic execution performance. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesBroadcastInterval	The length of time in seconds, between each broadcast attempt from the load balancer daemon.
nasKesApplogicBroadcastInterval	The length of time in seconds, between each broadcast of AppLogics load information across all the server in the cluster. This should be greater than nasKesBroacastInterval.
nasKesServerBroadcastInterval	The length of time in seconds, between each broadcast of server load information across all the server in the cluster. This should be greater than nasKesBroacastInterval.
nasKesServerLoadUpdateInterval	The length of time in seconds between each update of server load informations. A server load update applies the server load data that has been sampled up until the moment when the update occurs.
nasKesCpuLoadUpdateInterval	The length of time, in seconds, between each sampling of CPU usage.
nasKesDiskLoadUpdateInterval	The length of time, in seconds, between each sampling of disk usage.
nasKesMemLoadUpdateInterval	The length of time, in seconds, between each sampling of memory thrashes.
nasKesTotalReqsUpdateInterval	The length of time, in seconds, between each sampling of the number of requests.
nasKesMaxHops	The maximum number of times a request can be load-balanced to another server.
nasKesODBCReqMinThread	The minimum number of threads reserved to process asynchronous requests.
nasKesODBCReqMaxThread	The maximum number of threads reserved to process asynchronous requests.

Measurement	Description
nasKesODBCCacheMaxConns	The maximum number of connections opened between NAS and the database.
nasKesODBCCacheFreeSlots	The minimum number of cached connections established between NAS and the database.
nasKesODBCCacheTimeout	The time after which an idle connection is dropped.
nasKesODBCCacheInterval	The interval in seconds at which the cache cleaner will try to disconnect connections already idle for longer than the specified timeout.
nasKesODBCConnGiveupTime	Maximum time the driver will try to connect to the database.
nasKesODBCCacheDebug	Turns on the connection cache debug information.
nasKesODBCResultSetInitRows	The number of rows fetched at once from the database.
nasKesODBCResultSetMaxRows	The maximum number of rows the cached result set can contain.
nasKesODBCResultSetMaxSize	The maximum size of result set the driver will cache
nasKesODBCSqlDebug	Turns on SQL debug information.
nasKesODBCEnableParser	Turns on SQL parsing.
nasKesORCLReqMinThread	The minimum number of threads reserved to process asynchronous requests.
nasKesORCLReqMaxThread	The maximum number of threads reserved to process asynchronous requests.
nasKesORCLCacheMaxConns	The maximum number of connections opened between NAS and the database.
nasKesORCLCacheFreeSlots	The minimum number of cached connections established between NAS and the database.

Measurement	Description
nasKesORCLCacheTimeout	The time after which an idle connection is dropped.
nasKesORCLCacheInterval	The interval in seconds at which the cache cleaner will try to disconnect connections already idle for longer than the specified timeout.
nasKesORCLConnGiveupTime	The maximum time the driver will spend trying to obtain a connection to Oracle.
nasKesORCLCacheDebug	Turns on the connection cache debug information.
nasKesORCLResultSetInitRows	The number of rows fetched at once from the database.
nasKesORCLResultSetMaxRows	The maximum number of rows the cached result set can contain.
nasKesORCLResultSetMaxSize	The maximum size of result set the driver will cache.
nasKesORCLSqlDebug	Turns on SQL debug information.
nasKesSYBReqMinThread	The minimum number of threads reserved to process asynchronous requests.
nasKesSYBReqMaxThread	The maximum number of threads reserved to process asynchronous request.
nasKesSYBCacheMaxConns	The maximum number of connections opened between NAS and the database.
nasKesSYBCacheFreeSlots	The minimum number of cached connections established between NAS and the database.
nasKesSYBCacheTimeout	The time after which an idle connection is dropped.
nasKesSYBCacheInterval	The interval time between cached connections.
nasKesSYBConnGiveupTime	The maximum time the driver will spend trying to obtain a connection to Sybase before giving up.

Measurement	Description
nasKesSYBCacheDebug	Turns on the connection cache debug information.
nasKesSYBResultSetInitRows	The number of rows fetched at once from the database.
nasKesSYBResultSetMaxRows	The maximum number of rows the cached result set can contain.
nasKesSYBResultSetMaxSize	The maximum size of result set the driver will cache.

Engine Performance Counters

Measurement	Description
nasEngKesPort	The port of the KXS this engine serves. This is supplied as part of the object ID and cannot be modified after creation.
nasEngPort	The TCP/IP port this engine is listening on. The port can only be specified at the creation of the engine. It is not allowed to modify it.
nasEngType	Type of the engine: executive(0), Java(1000), C++(3000).
nasEngId	The ID is an incremental number starting at 0. The ID cannot be modified.
nasEngName	The name of this engine. This is an informational string that contains kcs, kxs ot kjs.
nasEngNewConsole	Starts each engine in a new console window.
nasEngStatus	The status column used to add, remove, enable or disable an engine. To create an engine, one needs to set. This follows rfc1443.
nasEngMinThread	The default minimum number of threads per engine.
nasEngMaxThread	The default maximum number of threads per engine.
nasEngReqRate	The rate at which requests arrive.

Measurement	Description
nasEngTotalReq	The total number of requests processed since engine startup.
nasEngReqNow	The number of requests being processed.
nasEngReqWait	The requests waiting to be serviced.
nasEngReqReady	The requests that are ready to be serviced.
nasEngAvgReqTime	The average request processing time.
nasEngThreadNow	Number of threads in use by the request manager.
nasEngThreadWait	The number of idle threads.
nasEngWebReqQueue	The number of web requests that are queued.
nasEngFailedReq	The number of requests that failed.
nasEngTotalConn	The total number of connections opened.
nasEngTotalConnNow	The total number of connections in use.
nasEngTotalAccept	The total number of connections listening to incoming requests.
nasEngTotalAcceptNow	The total number of connections listening to incoming connections in use.
nasEngTotalSent	The total number of packets sent.
nasEngTotalSentBytes	The total number of bytes sent.
nasEngTotalRecv	The total number of packets received.
nasEngTotalRecvBytes	The total number of bytes received.
nasEngBindTotal	The number of AppLogic bound since startup.
nasEngBindTotalCached	The number of AppLogic cached since startup.
nasEngTotalThreads	Total number of threads created in this process.

Measurement	Description
nasEngCurrentThreads	Total number of threads in use in this process.
nasEngSleepingThreads	Number of threads sleeping in this process.
nasEngDAETotalQuery	Total number of queries executed since startup.
nasEngDAEQueryNow	The number of queries being processed.
nasEngDAETotalConn	The number of logical connections created since startup.
nasEngDAEConnNow	The number of logical connections in use.
nasEngDAECacheCount	The number of caches.
nasEngODBCQueryTotal	Total number of queries executed since startup.
nasEngODBCPreparedQueryTotal	Total number of odbc prepared queries executed since startup.
nasEngODBCConnTotal	Total number of connections opened since startup.
nasEngODBCConnNow	Number of connections currently opened.
nasEngORCLQueryTotal	Total number of queries executed since startup.
nasEngORCLPreparedQueryTotal	Total number of prepared queries executed since startup.
nasEngORCLConnTotal	Total number of connections established with Oracle since startup.
nasEngORCLConnNow	Number of connections opened with Oracle now.
nasEngSYBQueryTotal	Total number of queries the driver processed since startup.

Measurement	Description
nasEngSYBPrepared QueryTotal	Total number of prepared queries processed since startup.
nasEngSYBConnTotal	Total number of connections opened since startup.
nasEngSYBConnNow	Number of SYB connections opened now.
nasStatusTrapEntry	The KES definition.
nasTrapKesIpAddress	The IP Address of KES host.
nasTrapKesPort	The port of the main engine of this NAS.
nasTrapEngPort	The port of the engine generating this event.
nasTrapEngState	The port of the engine generating this event.

Configuring the Microsoft Active Server Pages Monitor

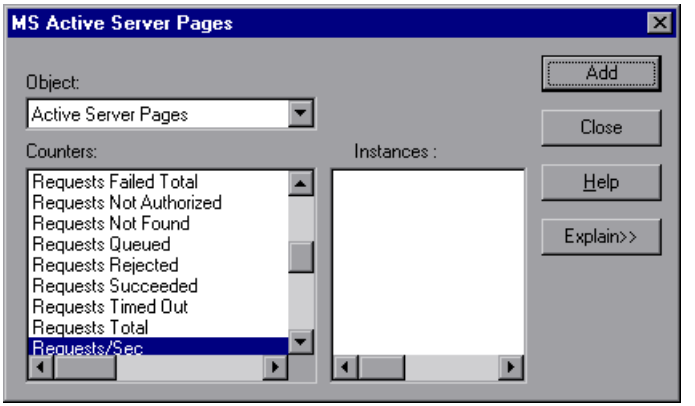
You select measurements to monitor the Microsoft ASP application server using the MS Active Server Pages dialog box.

Note: To monitor an ASP server through a firewall, use TCP, port 139.

To configure the ASP monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **MS Active Server Pages**, and then click **Add**.

The MS Active Server Pages dialog box opens, displaying the Active Server Pages object, its counters, and instances.



For a list of the available performance counters, see page 443.

Note: To change the default counters for the Microsoft ASP monitor, see "Changing a Monitor's Default Counters" on page 775.

---

5 Select a counter and instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and then click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

MS Active Server Pages Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Errors per Second	The number of errors per second.
Requests Wait Time	The number of milliseconds the most recent request was waiting in the queue.
Requests Executing	The number of requests currently executing.
Requests Queued	The number of requests waiting in the queue for service.
Requests Rejected	The total number of requests not executed because there were insufficient resources to process them.
Requests Not Found	The number of requests for files that were not found.
Requests/sec	The number of requests executed per second.
Memory Allocated	The total amount of memory, in bytes, currently allocated by Active Server Pages.
Errors During Script Run-Time	The number of failed requests due to run-time errors.
Sessions Current	The current number of sessions being serviced.
Transactions/sec	The number of transactions started per second.

Configuring the Oracle9iAS HTTP Monitor

You select measurements to monitor the Oracle9iAS HTTP server using the Oracle HTTP Server Monitor Configuration dialog box. Note that you must start running the Oracle9iAS HTTP server before you begin selecting the measurements you want to monitor.

Note: The port you use to monitor an Oracle9iAS HTTP server through a firewall depends on the configuration of your server.

To configure the Oracle9iAS HTTP monitor:

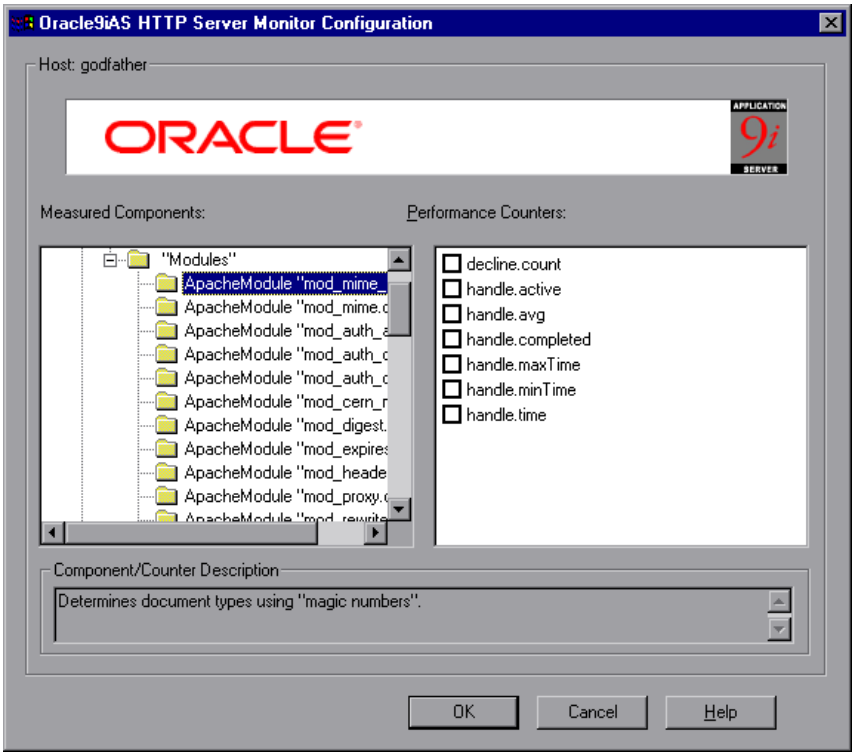
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **Oracle9iAS HTTP Server**, and then click **Add**.

The Oracle9iAS HTTP Server Monitor Configuration dialog box opens, displaying the counters that can be monitored.



5 Browse the Measured Components tree, and check the required counters and/or modules in the Oracle HTTP Server Monitor Configuration window's right pane.

For a list of the available performance counters, see page 446.

6 Click **OK** in the Oracle9iAS HTTP Server Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the Oracle9iAS HTTP monitor.

Oracle9iAS HTTP Server Performance Counters

The following table describes some of the modules that can be monitored:

Measurement	Description
mod_mime.c	Determines document types using file extensions
mod_mime_magic.c	Determines document types using "magic numbers"
mod_auth_anon.c	Provides anonymous user access to authenticated areas
mod_auth_dbm.c	Provides user authentication using DBM files
mod_auth_digest.c	Provides MD5 authentication
mod_cern_meta.c	Supports HTTP header metafiles
mod_digest.c	Provides MD5 authentication (deprecated by mod_auth_digest)
mod_expires.c	Applies Expires: headers to resources
mod_headers.c	Adds arbitrary HTTP headers to resources
mod_proxy.c	Provides caching proxy abilities
mod_rewrite.c	Provides powerful URI-to-filename mapping using regular expressions
mod_speling.c	Automatically corrects minor typos in URLs
mod_info.c	Provides server configuration information
mod_status.c	Displays server status
mod_usertrack.c	Provides user tracking using cookies
mod_dms.c	Provides access to DMS Apache statistics
mod_perl.c	Allows execution of perl scripts
mod_fastcgi.c	Supports CGI access to long-lived programs
mod_ssl.c	Provides SSL support
mod_plsql.c	Handles requests for Oracle stored procedures
mod_isapi.c	Provides Windows ISAPI extension support

Measurement	Description
mod_setenvif.c	Sets environment variables based on client information
mod_actions.c	Executes CGI scripts based on media type or request method
mod_imap.c	Handles imagemap files
mod_asis.c	Sends files that contain their own HTTP headers
mod_log_config.c	Provides user-configurable logging replacement for mod_log_common
mod_env.c	Passes environments to CGI scripts
mod_alias.c	Maps different parts of the host file system in the document tree, and redirects URLs
mod_userdir.c	Handles user home directories
mod_cgi.c	Invokes CGI scripts
mod_dir.c	Handles the basic directory
mod_autoindex.c	Provides automatic directory listings
mod_include.c	Provides server-parsed documents
mod_negotiation.c	Handles content negotiation
mod_auth.c	Provides user authentication using text files
mod_access.c	Provides access control based on the client hostname or IP address
mod_so.c	Supports loading modules (.so on UNIX, .dll on Win32) at run-time
mod_oprocmgr.c	Monitors JServ processes and restarts them if they fail
mod_jserv.c	Routes HTTP requests to JServ server processes. Balances load across multiple JServs by distributing new requests in round-robin order

Measurement	Description
mod_ose.c	Routes requests to the JVM embedded in Oracle's database server
http_core.c	Handles requests for static Web pages

The following table describes the counters that can be monitored:

Measurement	Description
handle.minTime	The minimum time spent in the module handler
handle.avg	The average time spent in the module handler
handle.active	The number of threads currently in the handle processing phase
handle.time	The total amount of time spent in the module handler
handle.completed	The number of times the handle processing phase was completed
request.maxTime	The maximum amount of time required to service an HTTP request
request.minTime	The minimum amount of time required to service an HTTP request
request.avg	The average amount of time required to service an HTTP request
request.active	The number of threads currently in the request processing phase
request.time	The total amount of time required to service an HTTP request
request.completed	The number of times the request processing phase was completed
connection.maxTime	The maximum amount of time spent servicing any HTTP connection
connection.minTime	The minimum amount of time spent servicing any HTTP connection

Measurement	Description
connection.avg	The average amount of time spent servicing HTTP connections
connection.active	The number of connections with currently open threads
connection.time	The total amount of time spent servicing HTTP connections
connection.completed	The number of times the connection processing phase was completed
numMods.value	The number of loaded modules
childFinish.count	The number of times the Apache parent server started a child server, for any reason
childStart.count	The number of times “children” finished “gracefully.”There are some ungraceful error/crash cases that are not counted in childFinish.count
Decline.count	The number of times each module declined HTTP requests
internalRedirect.count	The number of times that any module passed control to another module using an “internal redirect”
cpuTime.value	The total CPU time utilized by all processes on the Apache server (measured in CPU milliseconds)
heapSize.value	The total heap memory utilized by all processes on the Apache server (measured in kilobytes)
pid.value	The process identifier of the parent Apache process
upTime.value	The amount of time the server been running (measured in milliseconds)

Configuring the SilverStream Monitor

To monitor a SilverStream server you need to know the server statistics information URL. A simple way to verify the statistics URL is to access it from a browser.

The URL should be in the following format:

http://<*server_name/IP_address*>:<*port_number*>/SilverStream/Statistics

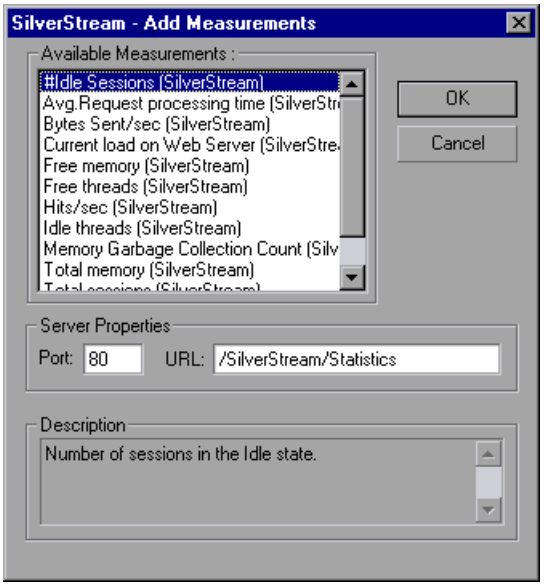
for example:

http://199.203.78.57:80/SilverStream/Statistics

To configure the SilverStream monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **SilverStream**, and then click **Add**.

A dialog box displaying the available measurements and server properties opens.



5 Select the required measurements. You can select multiple measurements using the **Ctrl** key.

For a list of the available performance counters, see page 452.

6 In the Server Properties section, enter the Port number and URL (without the server name), and click **OK**. The default URL is /SilverStream/Statistics.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: The default port number and URL can vary from one server to another. Please consult the Web server administrator.

To change the default server properties:

1 Open the *SilverStream.cfg* file in the *<LoadRunner Tuning Module root folder>\dat\ monitors* directory.

2 Edit the following parameters at the end of the file:

InfoURL	server statistics information URL
ServerPort	server port number
SamplingRate	rate (milliseconds) at which the LoadRunner Tuning Module monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner Tuning Module will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor a SilverStream server through a firewall, use the Web server port (by default, port 80).

SilverStream Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
#Idle Sessions	The number of sessions in the Idle state.
Avg. Request processing time	The average request processing time.
Bytes Sent/sec	The rate at which data bytes are sent from the Web server.
Current load on Web Server	The percentage of load utilized by the SilverStream server, scaled at a factor of 25.
Hits/sec	The HTTP request rate.
Total sessions	The total number of sessions.

Measurement	Description
Free memory	The total amount of memory in the Java Virtual Machine currently available for future allocated objects.
Total memory	The total amount of memory in the Java Virtual Machine.
Memory Garbage Collection Count	The total number of times the JAVA Garbage Collector has run since the server was started.
Free threads	The current number of threads not associated with a client connection and available for immediate use.
Idle threads	The number of threads associated with a client connection, but not currently handling a user request.
Total threads	The total number of client threads allocated.

Configuring the WebLogic (SNMP) Monitor

The WebLogic (SNMP) monitor uses SNMP to retrieve server statistics. To use this monitor, you must make sure that a version prior to WebLogic 8.0 is installed on your server, and that the SNMP agent is installed and activated on the server. For instructions on installing the SNMP agent, see http://www.weblogic.com/docs51/admindocs/snmpagent.html.

Note: To monitor a WebLogic (SNMP) server, use port 161 or 162, depending on the configuration of the agent.

To configure the WebLogic (SNMP) monitor:

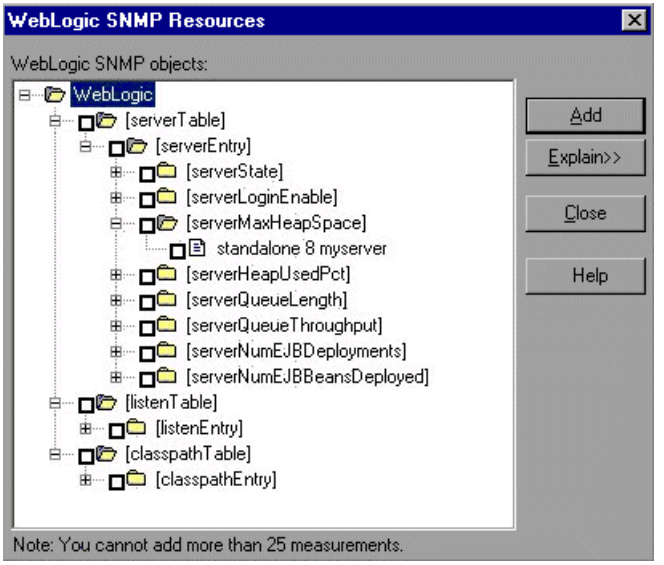
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebLogic**, and then click **Add**.

The WebLogic SNMP Resources dialog box displays the available measurements.



Note: You need to define the port number if the WebLogic SNMP agent is running on a different port than the default SNMP port. You can define the default port for your WebLogic server in the configuration file, *snmp.cfg*, located in *<LoadRunner Tuning Module root folder>\dat\monitors*. For example, if the port used by the SNMP agent on your WebLogic server is 8888, you should edit the *snmp.cfg* file as follows:

```
; WebLogic
[cm_snmp_mon_isp]
port=8888
```

Note: The WebLogic (SNMP) monitor can only monitor up to 25 measurements.

5 Browse the WebLogic SNMP Objects tree, and select the objects you want to measure.

For a list of the available performance counters, see page 455.

6 Click **Add** to place the selected objects on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

WebLogic (SNMP) Performance Counters

The following tables describe the measurements and server properties that can be monitored:

Server Table

The Server Table lists all WebLogic (SNMP) servers that are being monitored by the agent. A server must be contacted or be reported as a member of a cluster at least once before it will appear in this table. Servers are only reported as a member of a cluster when they are actively participating in the cluster, or shortly thereafter.

Measurement	Description
ServerState	The state of the WebLogic server, as inferred by the SNMP agent. **Up** implies that the agent can contact the server. **Down** implies that the agent cannot contact the server.
ServerLoginEnable	This value is true if client logins are enabled on the server.
ServerMaxHeapSpace	The maximum heap size for this server, in KB
ServerHeapUsedPct	The percentage of heap space currently in use on the server

Measurement	Description
ServerQueueLength	The current length of the server execute queue
ServerQueueThroughput	The current throughput of execute queue, expressed as the number of requests processed per second
ServerNumEJBDeployment	The total number of EJB deployment units known to the server
ServerNumEJBBeansDeployed	The total number of EJB beans actively deployed on the server

Listen Table

The Listen Table is the set of protocol, IP address, and port combinations on which servers are listening. There will be multiple entries for each server: one for each protocol, ipAddr, port combination. If clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
ListenPort	Port number.
ListenAdminOK	True if admin requests are allowed on this (protocol, ipAddr, port); otherwise false
ListenState	Listening if the (protocol, ipAddr, port) is enabled on the server; not Listening if it is not. The server may be listening but not accepting new clients if its server Login Enable state is false. In this case, existing clients will continue to function, but new ones will not.

ClassPath Table

The ClassPath Table is the table of classpath elements for Java, WebLogic (SNMP) server, and servlets. There are multiple entries in this table for each server. There may also be multiple entries for each path on a server. If

clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
CPType	The type of CP element: Java, WebLogic, servlet. A Java CPType means the cpElement is one of the elements in the normal Java classpath. A WebLogic CPType means the cpElement is one of the elements in weblogic.class.path. A servlet CPType means the cpElement is one of the elements in the dynamic servlet classpath.
CPIndex	The position of an element within its path. The index starts at 1.

Configuring the WebLogic (JMX) Monitor

The BEA WebLogic (JMX) monitor uses the Java JMX interface to access run-time MBeans on the server. An MBean is a container that holds the performance data.

Before using the WebLogic (JMX) monitor, you must install Java 1.3 or later on the Console machine. If Java 1.3 or later is already installed, but is not the default Java version being used, specify the full path to the updated version. You specify the path in the <LoadRunner Tuning Module *root folder*>*dat**monitors**WebLogicMon.ini* file. Edit the JVM entry in the [WebLogicMon] section. For example:

```
JVM="E:\Program Files\JavaSoft\JRE\1.3.1\bin\javaw.exe
```

Note: To use the WebLogic (JMX) monitor, you must make sure that WebLogic 6.0 or above is installed on your server.

Setting Permissions for Monitoring

You must set certain permissions for a user to be able to monitor MBeans.

To set permissions:

1. Open the WebLogic console (http://<host:port>/console).
2. In the tree on the left, select **Security** > **ACLs**.

   If you are working with the WebLogic 6.1 console, click **Create a new ACL...** in the screen on the right.
3. In the New ACL Name box, type weblogic.admin.mbean, and click **Create**.

   If you are working with the WebLogic 6.1 console, click **Add a new Permission...** in the screen on the right.
4. In the New Permission box (or Permission box, in the WebLogic 6.1 console), type access. In the WebLogic 6.0 console, click **Create**.
5. In the Users box and Groups box, enter the name of any user or group you want to use for monitoring.
6. Click **Grant Permission** in the WebLogic 6.0 console. In the WebLogic 6.1 console, click **Apply**.

Loading Classes from the Server

The WebLogic (JMX) monitor utilizes a built-in server called the ClasspathServlet to load classes directly and automatically from the server. The advantages of this are easy installation and version independence. The disadvantages are a slight decrease in performance when loading classes for the first time (due to the size of the servlet), and the possibility of the servlet becoming disabled.

If the servlet is disabled, or if you do not want to use the servlet, you can load classes directly from the file system.

To load classes directly from the file system:

1. Copy the *weblogic.jar* file from the application server install folder (under the lib folder) to *<LoadRunner Tuning Module root folder>\classes*.
2. If the classes file is not located in the default *<LoadRunner Tuning Module root folder>* folder, you need to specify the full path to it in the *<LoadRunner*

Tuning Module root folder>*dat**monitors**WebLogicMon.ini* file. In this file, change the line Weblogic=weblogic.jar to Weblogic=<*full path to weblogic.jar*>.

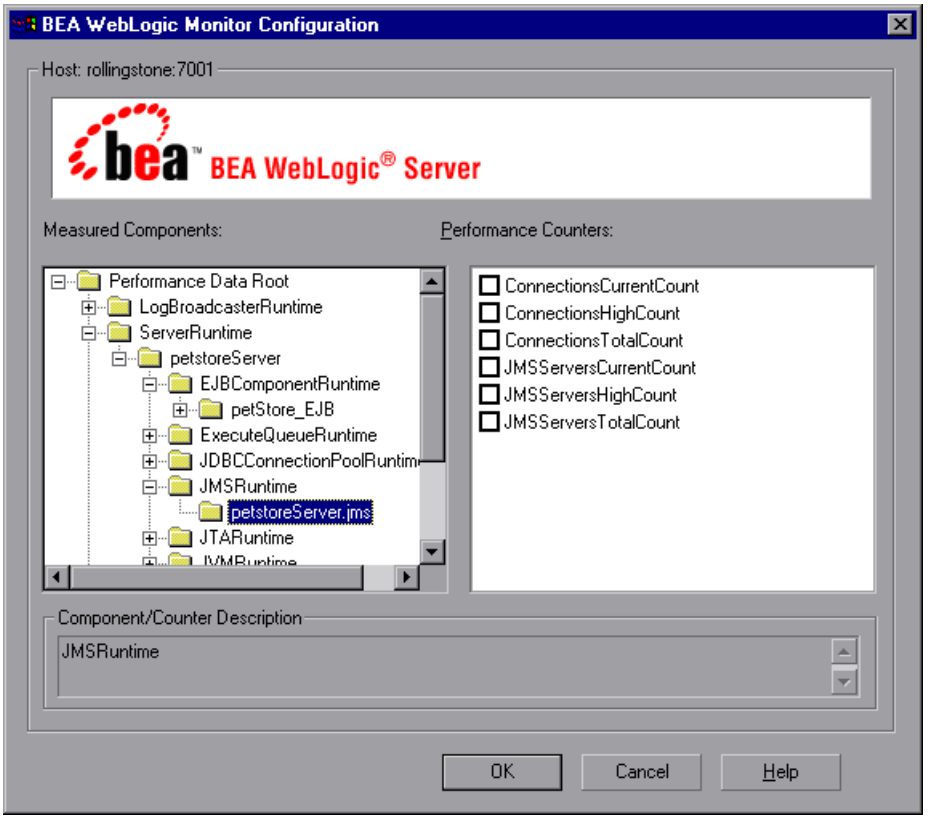
Configuring the WebLogic (JMX) Monitor

You select measurements to monitor the WebLogic (JMX) application server using the BEA WebLogic Monitor Configuration dialog box.

*LoadRunner Tuning Module***To configure the WebLogic (JMX) Monitor:**

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **WebLogic (JMX)**, and then click **Add**.
5. In the Enter Login Information dialog box, enter the username and password of a user with administrative privileges to the WebLogic server and click **OK**.

The BEA WebLogic Monitor Configuration dialog box opens.



For details on creating user permissions, see "Setting Permissions for Monitoring" on page 458.

6 Browse the Measured Components tree, and check the required performance counters in the BEA WebLogic Monitor Configuration window's right pane.

For a list of the available performance counters, see page 461.

7 Click **OK** in the BEA WebLogic Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the WebLogic (JMX) monitor.

BEA WebLogic (JMX) Performance Counters

The following measurements are available for the WebLogic (JMX) server:

LogBroadcasterRuntime

Measurement	Description
MessagesLogged	The number of total log messages generated by this instance of the WebLogic server.
Registered	Returns "false" if the MBean represented by this object has been unregistered.
CachingDisabled	Private property that disables caching in proxies.

ServerRuntime

For more information on the measurements contained in each of the following measurement categories, see Mercury Interactive's Load Testing Monitors Web site (http://www-heva.mercuryinteractive.com/products/loadrunner/load_testing_monitors/bealogic.html).

- ServletRuntime
- WebAppComponentRuntime
- EJBStatefulHomeRuntime
- JTARuntime
- JVMRuntime
- EJBEntityHomeRuntime.
- DomainRuntime
- EJBComponentRuntime
- DomainLogHandlerRuntime
- JDBCConnectionPoolRuntime
- ExecuteQueueRuntime
- ClusterRuntime

- JMSRuntime
- TimeServiceRuntime
- EJBStatelessHomeRuntime
- WLECConnectionServiceRuntime

ServerSecurityRuntime

Measurement	Description
UnlockedUsersTotalCount	Returns the number of times a user has been unlocked on the server
InvalidLoginUsersHighCount	Returns the high-water number of users with outstanding invalid login attempts for the server
LoginAttemptsWhileLockedTotalCount	Returns the cumulative number of invalid logins attempted on the server while the user was locked
Registered	Returns "false" if the MBean represented by this object has been unregistered.
LockedUsersCurrentCount	Returns the number of currently locked users on the server
CachingDisabled	Private property that disables caching in proxies.
InvalidLoginAttemptsTotalCount	Returns the cumulative number of invalid logins attempted on the server
UserLockoutTotalCount	Returns the cumulative number of user lockouts done on the server

Configuring the WebSphere Monitor

To monitor WebSphere version 5.x, you need to deploy the performance servlet on the application server using the IBM WebSphere "Installing a New Application" wizard.

To deploy the performance servlet on the application server for WebSphere 5.x:

1. From the administrative console, click **Applications** > **Install New Application** in the console navigation tree.
2. For Path, specify the full path name of the source application file ("PerfServletApp.ear") on the server machine and click **Next**.
3. Select the **Generate Default Bindings** check box and click **Next**.
4. On the Install New Application page, click **Summary**, and select the **Cell/Node/Server** option. Click **Click here**.
5. On the **Map modules to application servers** panel, select the server onto which the application files will install from the **Clusters and Servers** list, and select **Module** to select all of the application modules.
6. Click **Next**, and in the Summary panel click **Finish**.
7. Verify that the servlet is running properly and that the performance data is being generated. A simple way to verify that the performance data is accessible is to display it in a Web browser. The URL must be in the following format:

   http://<*server name*:*port number*>/<*servlet_folder*>/com.ibm.ivb.epm.servlet.PerformanceServlet

   For example: http://websphere.mercury.co.il:81/servlet/com.ibm.ivb.epm.servlet.PerformanceServlet

---

Note: Only browsers that are XML-compatible will allow you to view the performance XML file.

To configure the WebSphere monitor:

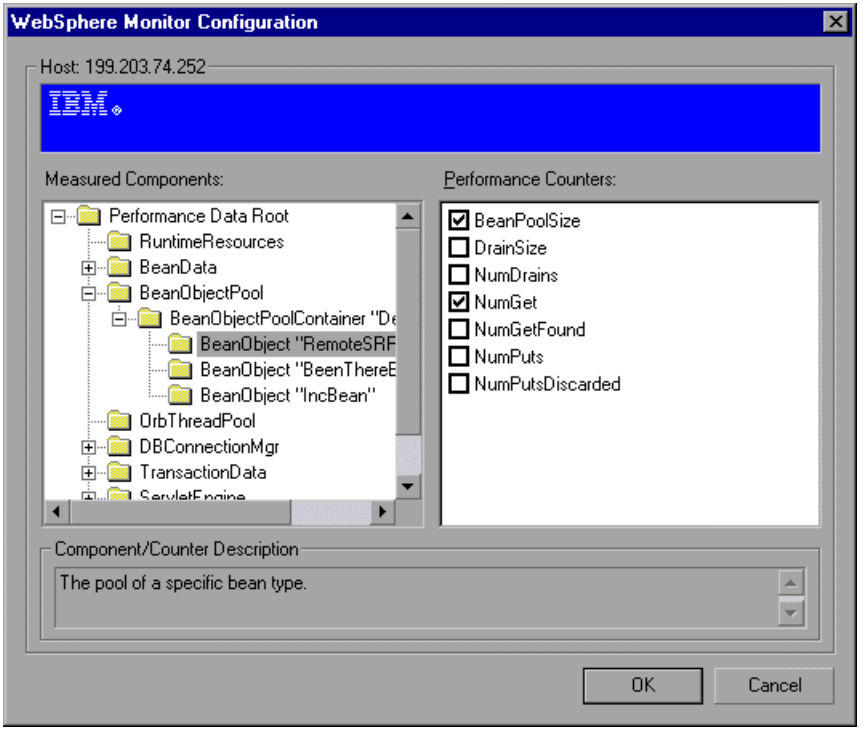
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebSphere**, and then click **Add**.

The WebSphere Monitor Configuration dialog box displays the available measurements.



5 Browse the Measured Components tree, and check the required performance counters in the WebSphere Monitor Configuration window's right pane.

For a list of the available performance counters, see page 467.

6 Click **OK** in the WebSphere Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the WebSphere monitor.

Note: The port you use to monitor a WebSphere server through a firewall depends on the configuration of your server.

To specify another Web alias for the servlet directory:

By default, LoadRunner Tuning Module uses the alias servlet as the servlet directory Web alias. For example, if the WebSphere Server machine is named mercury and the path for the servlets directory is: E:\AppServer\hosts\default_host\default_app\servlets, LoadRunner Tuning Module will request the XML file in the following URL: http:/mercury/servlet/com.ibm.ivb.epm.servlet.PerformanceServlet, where servlet is the default web alias for the servlet directory.

If the Web alias for the servlet directory is not servlet, you must specify the servlet directory Web alias in the Add Machine dialog box according to the following format:

http://<*server name:port number*>/<*servlet_dir_alias*>

For example: http://mercury/servlet2

Using this method, you can monitor as many application servers as you want—whether they are installed on the same machine, or on different machines.

To monitor other applications, in addition to the default application:

You can monitor as many applications as you want, regardless of whether they are installed on the same machine or different machines.

1 Copy the same files that you copied to the Servlets directory for the Default application to the Servlets directories for any other Web applications that you want to monitor.

2 Add the com.ibm.ivb.epm.servlet.PerformanceServlet to the configuration in the WebSphere Console for each Web application.

3 Add the Web application to be monitored to the WebSphere Performance Monitor using the following format:

http://*<server:port_number>/<servlet_dir_alias>/servlet*

For example: http://mercury/servlet3/servlet

To work with WebSphere version 3.5.x

1 The EPM counters in 3.5.x are by default set to "none". To enable the counters, choose the application server you are monitoring in the WebSphere Administrator's Console browser.

2 Right-click the application server and select **Performance**. Select Performance Modules from the pop-up window.

3 Right-click Performance Modules to choose a performance level. Selecting various levels of counters enables the application server to manage varying levels of performance data.

4 Click the **Set** button.

5 In versions 3.5.2 and 3.5.3 the Servlet counters have been disabled. To enable the Servlet counters, you need to modify the contents of the com/ibm/servlet/appserver.properties file located in "<WAS_HOME>\lib\ibmwebas.jar".

Extract the *jar* file and modify the appserver.properties as follows:

```
#listeners.application=com.ibm.servlet.engine.EPMApplicationListener
com.ibm.servlet.debug.OLTServletManager
listeners.application=
```

Should be:

```
listeners.application=com.ibm.servlet.engine.EPMApplicationListener
com.ibm.servlet.debug.OLTServletManager
#listeners.application=
```

6 Repackage the *jar* file.

WebSphere Performance Counters

The following tables describe the counters that can be monitored:

Run-Time Resources

Contains resources related to the Java Virtual Machine run-time, as well as the ORB.

Measurement	Description
MemoryFree	The amount of free memory remaining in the Java Virtual Machine
MemoryTotal	The total memory allocated for the Java Virtual Machine
MemoryUse	The total memory in use within the Java Virtual Machine

BeanData

Every home on the server provides performance data, depending upon the type of bean deployed in the home. The top level bean data holds an aggregate of all the containers.

Measurement	Description
BeanCreates	The number of beans created. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanCreates	The number of entity beans created
BeanRemoves	The number of entity beans pertaining to a specific bean that have been removed. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanRemoves	The number of entity beans removed
StatefulBeanCreates	The number of stateful beans created
StatefulBeanRemoves	The number of stateful bean removed

Measurement	Description
BeanPassivates	The number of bean passivates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanPassivates	The number of entity bean passivates
StatefulBeanPassivates	The number of stateful bean passivates
BeanActivates	The number of bean activates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanActivates	The number of entity bean activates
StatefulBeanActivates	The number of stateful bean activates
BeanLoads	The number of times the bean data was loaded. Applies to entity
BeanStores	The number of times the bean data was stored in the database. Applies to entity
BeanInstantiates	The number of times a bean object was created. This applies to an individual bean, regardless of its type.
StatelessBeanInstantiates	The number of times a stateless session bean object was created
StatefulBeanInstantiates	The number of times a stateful session bean object was created
EntityBeanInstantiates	The number of times an entity bean object was created
BeanDestroys	The number of times an individual bean object was destroyed. This applies to any bean, regardless of its type
StatelessBeanDestroys	The number of times a stateless session bean object was destroyed
StatefulBeanDestroys	The number of times a stateful session bean object was destroyed

Measurement	Description
EntityBeanDestroys	The number of times an entity bean object was destroyed
BeansActive	The average number of instances of active beans pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeansActive	The average number of active entity beans
StatefulBeansActive	The average number of active session beans
BeansLive	The average number of bean objects of this specific type that are instantiated but not yet destroyed. This applies to an individual bean, regardless of its type.
StatelessBeansLive	The average number of stateless session bean objects that are instantiated but not yet destroyed
StatefulBeansLive	The average number of stateful session bean objects that are instantiated but not yet destroyed
EntityBeansLive	The average number of entity bean objects that are instantiated but not yet destroyed
BeanMethodRT	The average method response time for all methods defined in the remote interface to this bean. Applies to all beans
BeanMethodActive	The average number of methods being processed concurrently. Applies to all beans
BeanMethodCalls	The total number of method calls against this bean

BeanObjectPool

The server holds a cache of bean objects. Each home has a cache and there is therefore one BeanObjectPoolContainer per container. The top level BeanObjectPool holds an aggregate of all the containers data.

Measurement	Description
BeanObjectPoolContainer	The pool of a specific bean type
BeanObject	The pool specific to a home
NumGet	The number of calls retrieving an object from the pool
NumGetFound	The number of calls to the pool that resulted in finding an available bean
NumPuts	The number of beans that were released to the pool
NumPutsDiscarded	The number of times releasing a bean to the pool resulted in the bean being discarded because the pool was full
NumDrains	The number of times the daemon found the pool was idle and attempted to clean it
DrainSize	The average number of beans discarded by the daemon during a clean
BeanPoolSize	The average number of beans in the pool

OrbThreadPool

These are resources related to the ORB thread pool that is on the server.

Measurement	Description
ActiveThreads	The average number of active threads in the pool
TotalThreads	The average number of threads in the pool
PercentTimeMaxed	The average percent of the time that the number of threads in the pool reached or exceeded the desired maximum number
ThreadCreates	The number of threads created

Measurement	Description
ThreadDestroys	The number of threads destroyed
ConfiguredMaxSize	The configured maximum number of pooled threads

DBConnectionMgr

These are resources related to the database connection manager. The manager consists of a series of data sources, as well as a top-level aggregate of each of the performance metrics.

Measurement	Description
DataSource	Resources related to a specific data source specified by the "name" attribute
ConnectionCreates	The number of connections created
ConnectionDestroys	The number of connections released
ConnectionPoolSize	The average size of the pool, i.e., number of connections
ConnectionAllocates	The number of times a connection was allocated
ConnectionWaiters	The average number of threads waiting for a connection
ConnectionWaitTime	The average time, in seconds, of a connection grant
ConnectionTime	The average time, in seconds, that a connection is in use
ConnectionPercentUsed	The average percentage of the pool that is in use
ConnectionPercentMaxed	The percentage of the time that all connections are in use

TransactionData

These are resources that pertain to transactions.

Measurement	Description
NumTransactions	The number of transactions processed
ActiveTransactions	The average number of active transactions
TransactionRT	The average duration of each transaction
BeanObjectCount	The average number of bean object pools involved in a transaction
RolledBack	The number of transactions rolled back
Commited	The number of transactions committed
LocalTransactions	The number of transactions that were local
TransactionMethodCount	The average number of methods invoked as part of each transaction
Timeouts	The number of transactions that timed out due to inactivity timeouts
TransactionSuspended	The average number of times that a transaction was suspended

ServletEngine

These are resources that are related to servlets and JSPs.

Measurement	Description
ServletsLoaded	The number of servlets currently loaded
ServletRequests	The number of requests serviced
CurrentRequests	The number of requests currently being serviced
ServletRT	The average response time for each request
ServletsActive	The average number of servlets actively processing requests

Measurement	Description
ServletIdle	The amount of time that the server has been idle (i.e., time since last request)
ServletErrors	The number of requests that resulted in an error or an exception
ServletBeanCalls	The number of bean method invocations that were made by the servlet
ServletBeanCreates	The number of bean references that were made by the servlet
ServletDBCalls	The number of database calls made by the servlet
ServletDBConAlloc	The number of database connections allocated by the servlet
SessionLoads	The number of times the servlet session data was read from the database
SessionStores	The number of times the servlet session data was stored in the database
SessionSize	The average size, in bytes, of a session data
LoadedSince	The time that has passed since the server was loaded (UNC time)

Sessions

These are general metrics regarding the HTTP session pool.

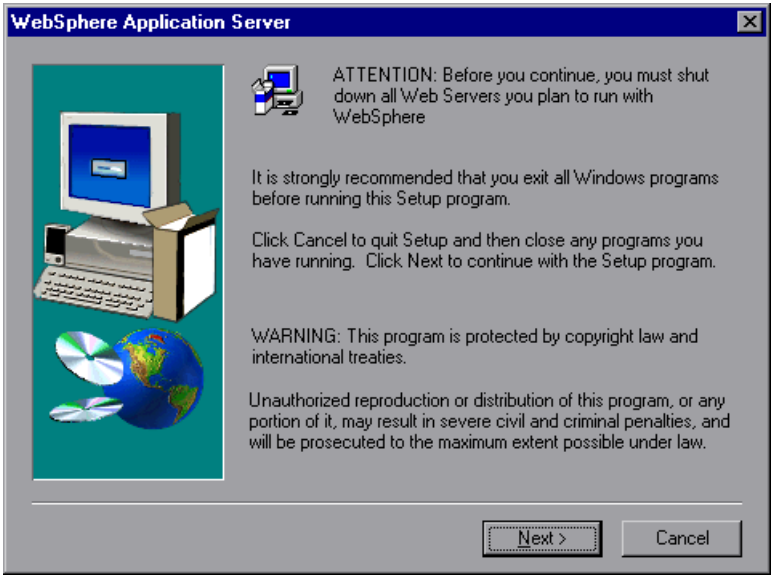
Measurement	Description
SessionsCreated	The number of sessions created on the server
SessionsActive	The number of currently active sessions
SessionsInvalidated	The number of invalidated sessions. May not be valid when using sessions in the database mode
SessionLifetime	Contains statistical data of sessions that have been invalidated. Does not include sessions that are still alive

Configuring the WebSphere (EPM) Monitor

To monitor the IBM WebSphere application server (3.5.x), you must first install the IBM WebSphere Administrator's Console on the Console machine. You may also need to copy the security keyring.

To install the IBM WebSphere Administrator's Console:

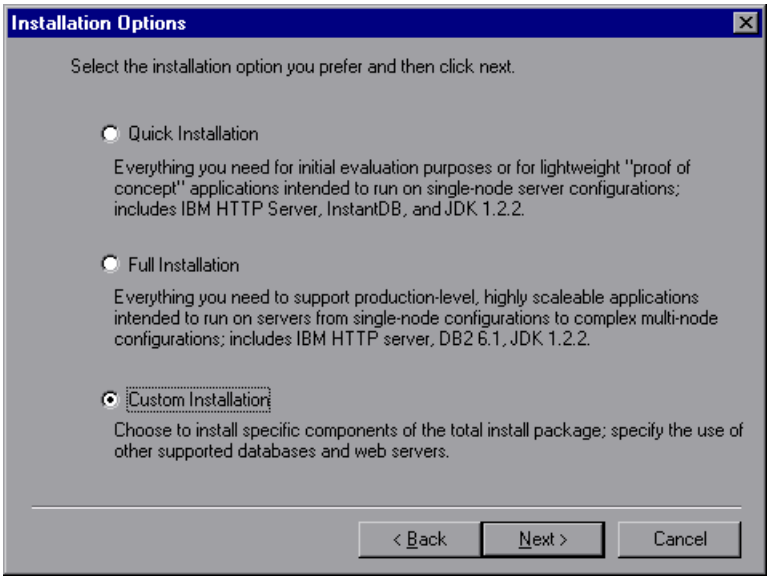
1 Start the WebSphere installation program from the WebSphere 3.5 Windows NT distribution CD-ROM. The WebSphere Application Server dialog box opens.



2 Disregard the instruction to shut down all Web servers that you plan to run with WebSphere. This is not relevant to the Administrator's Console installation.

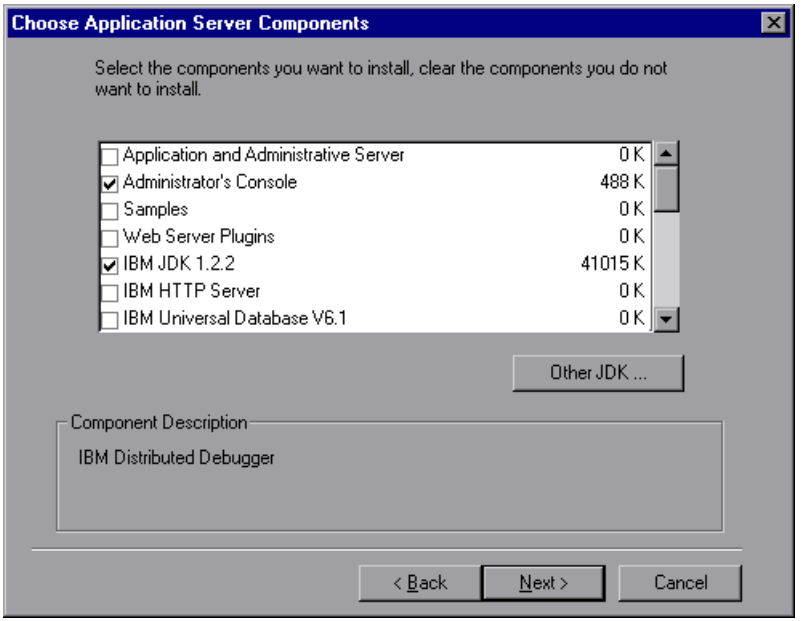
Click **Next**.

3 The Installation Options dialog box opens. Select **Custom Installation**.



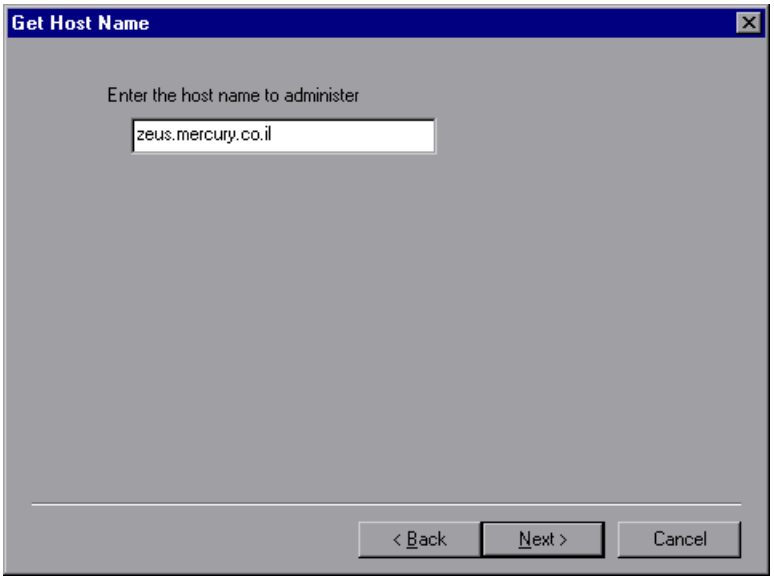
Click **Next**.

4 The Choose Application Server Components dialog box opens. Select **Administrator's Console** and **IBM JDK 1.2.2**. Clear all the other options.



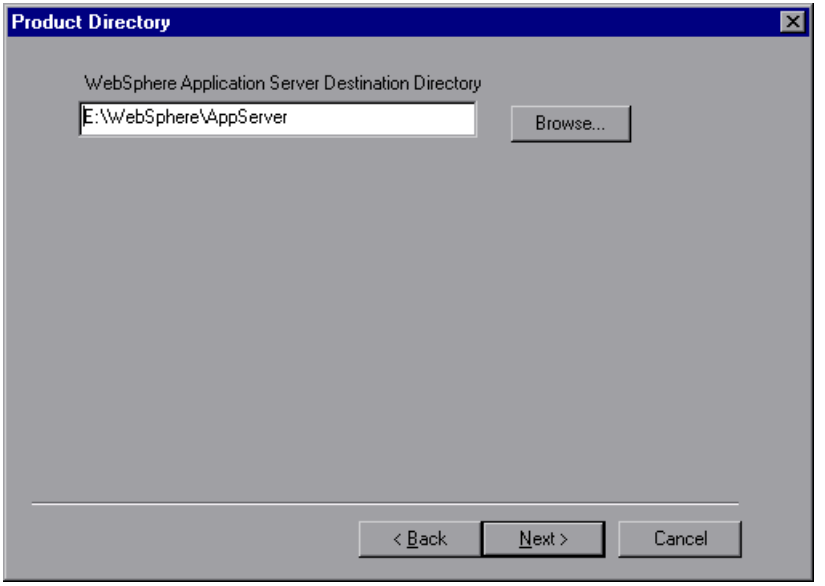
Click **Next**.

5 The Get Host Name dialog box opens. Type the name of the machine that you want to monitor.



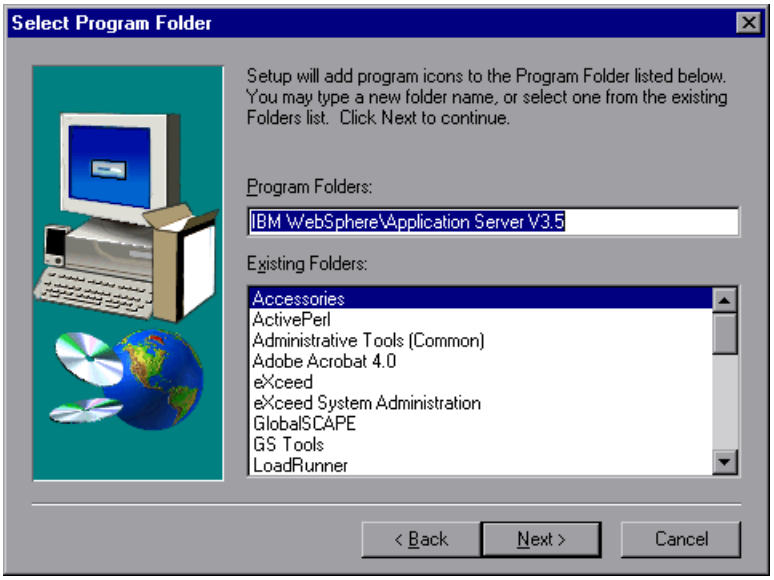
Click **Next**.

6 The Product Directory dialog box opens. Specify the folder in which to install the Administrator's Console. To select a different location, click **Browse**, choose a folder other than the default folder, and click **OK**.



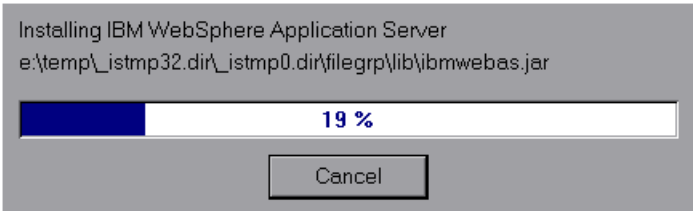
Click **Next**.

7 The Select Program Folder dialog box opens. Specify a program folder, or accept the default folder, IBM WebSphere\Application Server V3.5.

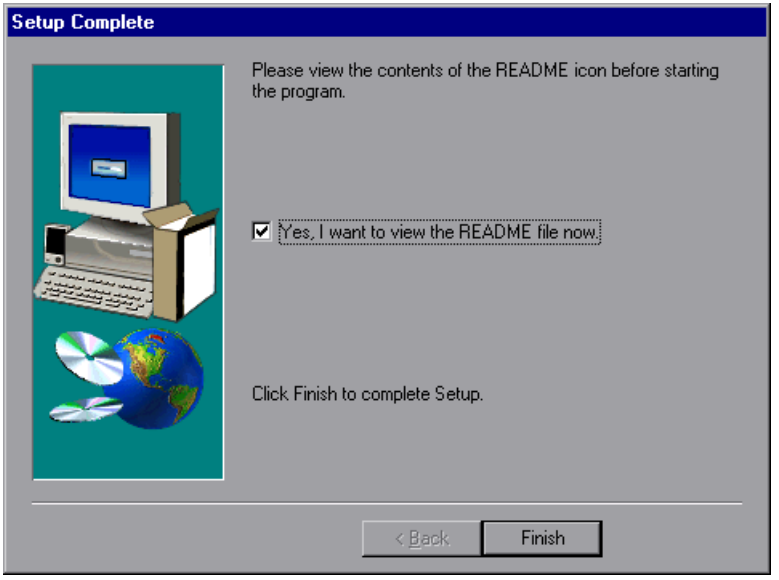


Click **Next**.

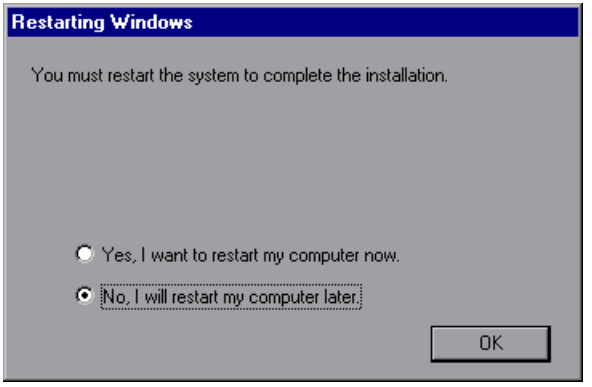
The installation process begins. To pause or quit the installation, click **Cancel**.



8 When the installation is complete, the Setup Complete dialog box opens. Select the check box to view the readme file before starting the program. You can view the readme file at any time by selecting **Start** > **Programs** > **Application Server V3.5** > **IBM WebSphere** > **README**.



9 Click **Finish** to complete the installation program. The Restarting Windows dialog box opens.



10 Select either to restart your computer and complete the installation now (recommended) or to wait and complete the installation later.

11 Click **OK** to complete the installation of the Administrator's Console.

Copying the Security Keyring

If you enabled security on the WebSphere server, you must copy the security keyring from the server to the admin client. (One way to tell whether security is enabled is to see whether the Administrator's Console can connect to the admin server.) A keyring is a certification used by the server to identify the client.

You need to copy the *jar* file containing the keyring from the server lib folder to the client lib folder. You also need to add the *jar* file containing the keyring to the monitoring client command line.

Note: The keyring used in this file (*353Keyring.jar*) is the IBM dummy keyring that must be installed on servers using versions 3.52 and below. If your server is using the IBM dummy keyring and is version 3.52 or below, you do not need to change the line. If you are using the dummy keyring and are running version 3.53 or later, you do not need to do anything.

To copy the keyring:

1 Copy the keyring *jar* file from the server to the admin client lib folder (by default, C:\Websphere\Appserver\lib):

The *jar* file containing the keyring, *xxxKeyring.jar*, is located by default in the following location:

NT Server	C:\Websphere\Appserver\lib
UNIX Server	OPT/websphere/Appserver/lib

2 Open the *<LoadRunner Tuning Module root folder>\dat\monitors\WebSphere35Mon.ini* file in a text editor.

3 Locate the following line:
JVM_CLASSES4=C:\WebSphere\AppServer\lib\353Keyring.jar

Note: If you did not use the default location for the WebSphere installation, the line will be different.

---

4 Change *353Keyring.jar* to the keyring you are using.

Enabling EPM Counters on the WebSphere 3.5.x Server

To enable the EPM counters, which are by default set to "none," right-click the application you are monitoring in the WebSphere Administrator's Console browser, and select **Performance**. Expand the Performance Modules tree in the dialog box that opens. In order to manage different levels of performance data, right-click the performance modules and choose a performance level. Click the **Set** button.

Alternatively, ensure that the application server is started, select the **Advanced** tab in the WebSphere Administrator's Console browser, and in the EPM Specification box, type:
epm=high:epm.beanMethodData=none

Configuring the WebSphere (EPM) Monitor

Once you have installed the WebSphere Administrator's Console and enabled the EPM counters, you can activate the WebSphere (EPM) monitor.

To activate the WebSphere EPM monitor:

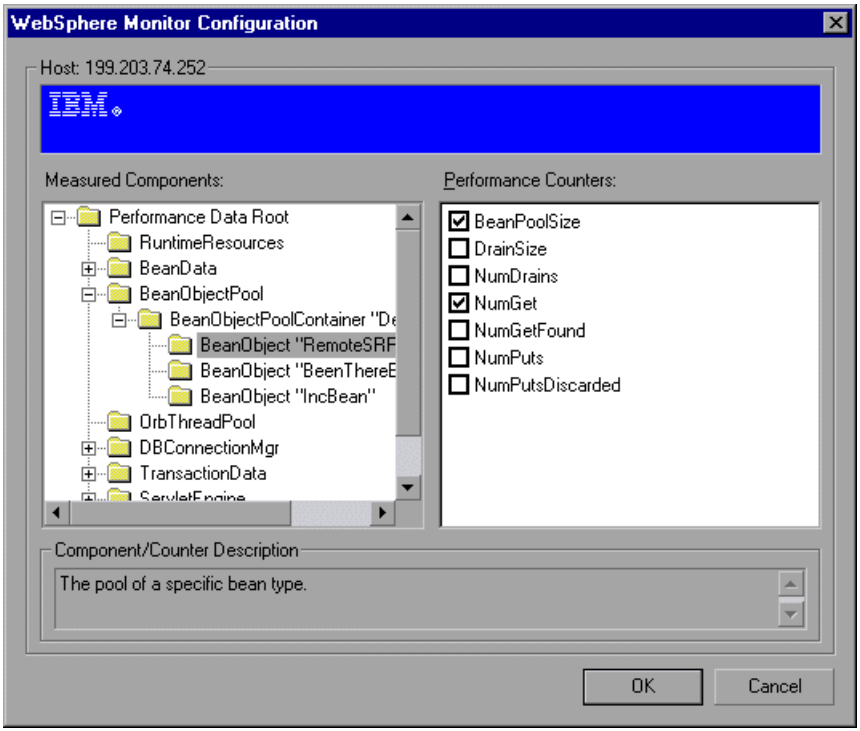
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebSphere (EPM)**, and then click **Add**.

The WebSphere Monitor Configuration dialog box displays the available measurements.



5 Browse the Measured Components tree, and check the required performance counters in the WebSphere Monitor Configuration window's right pane.

For a list of the available performance counters, see page 467.

6 Click **OK** in the WebSphere Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the WebSphere (EPM) monitor.

34

Database Resource Monitoring

You can monitor DB2, Oracle, SQL Server, or Sybase database resource usage during a session step run using LoadRunner Tuning Module's Database Server Resource monitors.

This chapter describes:

- Configuring the DB2 Monitor
- Configuring the Oracle Monitor
- Configuring the SQL Server Monitor
- Configuring the Sybase Monitor

About Database Resource Monitoring

The DB2, Oracle, SQL Server, or Sybase database server resource monitors measure statistics for DB2, Oracle, SQL Server, or Sybase database servers. During a session step run, you use these monitors to isolate database server performance bottlenecks.

For each database server, you configure the measurements you want to monitor before running your session step. Note that in order to run the DB2, Oracle, and Sybase monitors, you must also install the client libraries on the database server you want to monitor.

Configuring the DB2 Monitor

The DB2 database server monitor measures the resource usage on a DB2 database during a session step run.

Note: If there is no application working with a database, you can only monitor the database manager instance.

Before you can monitor a DB2 database server, you must set up the DB2 monitor environment.

To set up the DB2 monitor environment:

1. Install all the client files and libraries on the Console machine.
2. Select **Start** > **Programs** > **DB2 for Windows NT** > **Control Center**. Enter your DB2 server username and password (with administrative privileges).
3. In the console that opens, right-click **Systems**, and select **Add**.
4. Enter the following settings in the dialog box:

   System Name: <*server name*>

   Remote Instance: DB2

   Host Name: <*server name*>

   Service Name: the DB2 server port. The default value is 50000.

5. Click **Retrieve**, and then **OK**.

Note: If you receive an error message after clicking **Retrieve**, repeat steps 3 and 4, and click **OK**.

---

6. Expand the <*server name*> node in the console tree.
7. Right-click **Instance**, and select **Add**.

8 Enter the following settings in the dialog box:

Remote Instance: DB2

Instance Name: the database instance to be called from the Console

Host Name: <*server name*>

Service Name: the DB2 server port. The default value is 50000.

9 Click **OK** and close the Control Center.

Note: You can only work with a single Database Manager instance during each monitoring session.

To configure the DB2 monitor:

1 Click **Monitors** to open the Monitors Configuration dialog box.

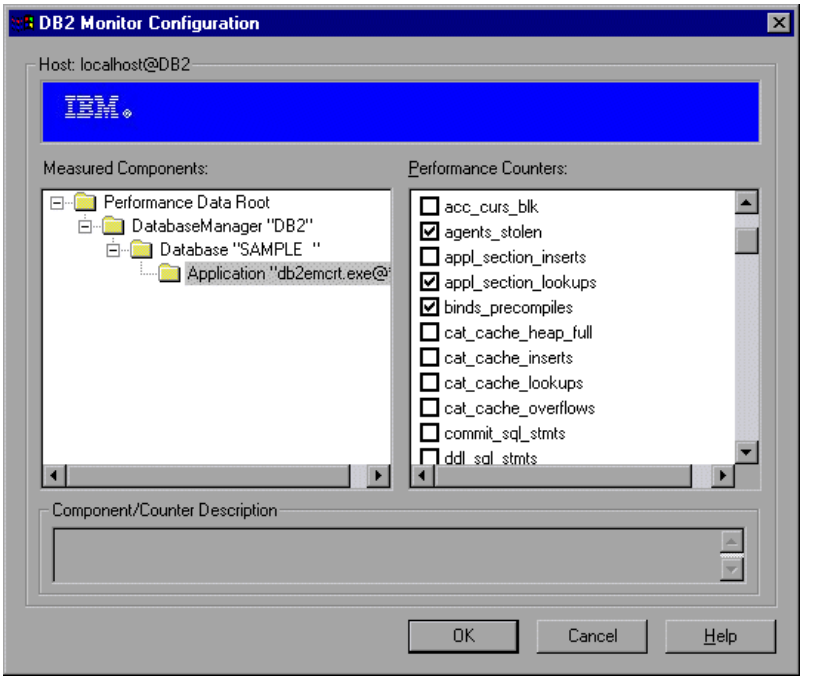
2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Database Server Resource category, select **DB2**, and then click **Add**.

5 In the dialog box that opens, enter your DB2 server username and password, and click **OK**.

The DB2 Monitor Configuration dialog box opens.



6 Expand the Measured Components tree and select the methods and counters you want to monitor.

For a list of the available performance counters, see **page 488**.

7 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

8 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

DB2 Performance Counters

The following tables describe the default counters that can be monitored.

DatabaseManager

Measurement	Description
rem_cons_in	The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.
rem_cons_in_exec	The number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.
local_cons	The number of local applications that are currently connected to a database within the database manager instance being monitored.
local_cons_in_exec	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
con_local_dbases	The number of local databases that have applications connected.
agents_registered	The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).
agents_waiting_on_token	The number of agents waiting for a token so they can execute a transaction in the database manager.
idle_agents	The number of agents in the agent pool that are currently unassigned to an application and are therefore "idle".
agents_from_pool	The number of agents assigned from the agent pool
agents_created_empty_pool	The number of agents created because the agent pool was empty.

Measurement	Description
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
comm_private_mem	The amount of private memory that the instance of the database manager has currently committed at the time of the snapshot.
inactive_gw_agents	The number of DRDA agents in the DRDA connections pool that are primed with a connection to a DRDA database, but are inactive.
num_gw_conn_switches	The number of times that an agent from the agents pool was primed with a connection and was stolen for use with a different DRDA database.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
post_threshold_sorts	The number of sorts that have requested heaps after the sort heap threshold has been reached.
piped_sorts_requested	The number of piped sorts that have been requested.
piped_sorts_accepted	The number of piped sorts that have been accepted.

Database

Measurement	Description
appls_cur_cons	Indicates the number of applications that are currently connected to the database.
appls_in_db2	Indicates the number of applications that are currently connected to the database, and for which the database manager is currently processing a request.

Measurement	Description
total_sec_cons	The number of connections made by a sub-agent to the database at the node.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
active_sorts	The number of sorts in the database that currently have a sort heap allocated.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	Indicates the number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.

Measurement	Description
pool_data_writes	Indicates the number of times a buffer pool data page was physically written to disk.
pool_index_l_reads	Indicates the number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	Indicates the number of physical read requests to get index pages into the buffer pool.
pool_index_writes	Indicates the number of times a buffer pool index page was physically written to disk.
pool_read_time	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
pool_write_time	Provides the total amount of time spent physically writing data or index pages from the buffer pool to disk.
files_closed	The total number of database files closed.
pool_async_data_reads	The number of pages read asynchronously into the buffer pool.
pool_async_data_writes	The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.
pool_async_index_writes	The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.
pool_async_index_reads	The number of index pages read asynchronously into the buffer pool by a pre-fetcher.
pool_async_read_time	The total elapsed time spent reading by database manager pre-fetchers.

Measurement	Description
pool_async_write_time	The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
pool_async_data_read_reqs	The number of asynchronous read requests.
pool_lsn_gap_clns	The number of times a page cleaner was invoked because the logging space used had reached a pre-defined criterion for the database.
pool_drty_pg_steal_clns	The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
pool_drty_pg_thrsh_clns	The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.
direct_reads	The number of read operations that do not use the buffer pool.
direct_writes	The number of write operations that do not use the buffer pool.
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.

Measurement	Description
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
pkg_cache_num_overflows	The number of times that the package cache overflowed the bounds of its allocated memory.
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
sec_logs_allocated	The total number of secondary log files that are currently being used for the database.

Measurement	Description
log_reads	The number of log pages read from disk by the logger.
log_writes	The number of log pages written to disk by the logger.
total_log_used	The total amount of active log space currently used (in bytes) in the database.
locks_held	The number of locks currently held.
lock_list_in_use	The total amount of lock list memory (in bytes) that is in use.
deadlocks	The total number of deadlocks that have occurred.
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	Indicates the number of agents waiting on a lock.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.

Measurement	Description
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.

Measurement	Description
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

Application

Measurement	Description
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space

Measurement	Description
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	Indicates the number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.
pool_data_writes	Indicates the number of times a buffer pool data page was physically written to disk.
pool_index_l_reads	Indicates the number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	Indicates the number of physical read requests to get index pages into the buffer pool.
pool_index_writes	Indicates the number of times a buffer pool index page was physically written to disk.
pool_read_time	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.

Measurement	Description
direct_reads	The number of read operations that do not use the buffer pool.
direct_writes	The number of write operations that do not use the buffer pool.
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.

Measurement	Description
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
uow_log_space_used	The amount of log space (in bytes) used in the current unit of work of the monitored application.
locks_held	The number of locks currently held.
deadlocks	The total number of deadlocks that have occurred.
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	Indicates the number of agents waiting on a lock.
uow_lock_wait_time	The total amount of elapsed time this unit of work has spent waiting for locks.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.

Measurement	Description
rows_written	The number of rows changed (inserted, deleted or updated) in the table.
rows_read	The number of rows read from the table.
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.
open_rem_curs	The number of remote cursors currently open for this application, including those cursors counted by 'open_rem_curs_blk'.
open_rem_curs_blk	The number of remote blocking cursors currently open for this application.
rej_curs_blk	The number of times that a request for an I/O block at server was rejected and the request was converted to non-blocked I/O.
acc_curs_blk	The number of times that a request for an I/O block was accepted.
open_loc_curs	The number of local cursors currently open for this application, including those cursors counted by 'open_loc_curs_blk'.
open_loc_curs_blk	The number of local blocking cursors currently open for this application.
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.

Measurement	Description
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

Note: The port you use to monitor an Oracle server through a firewall depends on the configuration of the Oracle server. Configuration information for the connection between the client and server is located in the Oracle client **tnsnames.ora** file.

Configuring the Oracle Monitor

The Oracle server measures information from the V$SESSTAT and V$SYSSTAT Oracle V$ tables, and other table counters defined by the user in the custom query. In order to monitor the Oracle server, you must set up the monitoring environment as described below.

Note: The port you use to monitor an Oracle server through a firewall depends on the configuration of the Oracle server. Configuration information for the connection between the client and server is located in the Oracle client **tnsnames.ora** file.

To set up the Oracle monitor environment:

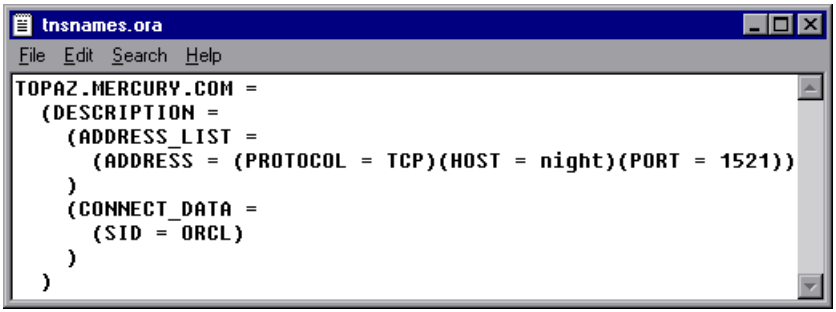
1. Ensure that the Oracle client libraries are installed on the Console machine.
2. Verify that *%OracleHome%\bin* is included in the path environment variable. If it is not, add it.
3. Configure the *tnsnames.ora* file on the Console machine so that the Oracle client can communicate with the Oracle server(s) you plan to monitor.

   You can configure connection parameters either manually, by editing the *tnsnames.ora* file in a text editor, or using the Oracle service configuration tool (for example, select **Start** > **Programs** > **Oracle for Windows NT** > **Oracle Net8 Easy Config**).

   You specify:

   - a new service name (TNS name) for the Oracle instance
   - TCP protocol
   - the host name (name of monitored server machine)
   - the port number (usually 1521)
   - the database SID (the default SID is ORCL)

For example:



Note: Only the 32-bit Oracle client should be installed on the Console machine running the Oracle monitor. If you have a 16-bit and a 32-bit Oracle client installation on the Console machine, the 16-bit installation should be uninstalled.

---

4. Obtain a username and password for the service from your database administrator, and ensure that the Console has database administrator privileges for the Oracle V$ tables (V$SESSTAT, V$SYSSTAT, V$STATNAME, V$INSTANCE, V$SESSION).
5. Verify connection with the Oracle server by performing *tns ping* from the Console machine. Note that there may be a problem connecting if the Oracle server is behind a DMZ/firewall that limits its communication to application servers accessing it.
6. Ensure that the registries are updated for the version of Oracle that you are using and that they have the following key:
   HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
7. Verify that the Oracle server you want to monitor is up and running.

Note: It is possible to monitor several Oracle database servers concurrently.

---

8 Run SQL*Plus from the Console and attempt to log in to the Oracle server(s) with the desired username/password/server combination.

9 Type SELECT * FROM V$SYSSTAT to verify that you can view the V$SYSSTAT table on the Oracle server. Use similar queries to verify that you can view the V$SESSTAT, V$SESSION, V$INSTANCE, V$STATNAME, and V$PROCESS tables on the server. Make sure that the Oracle bin directory is in the search path.

10 To change the length of each monitoring sample (in seconds), you need to edit the *dat\monitors\vmon.cfg* file in the LoadRunner Tuning Module root folder. The default rate is 10 seconds.

Note: The minimum sampling rate for the Oracle Monitor is 10 seconds. If you set the sampling rate at less than 10 seconds, the Oracle Monitor will continue to monitor at 10 second intervals.

Note: If a problem occurs in setting up the Oracle environment, view the error message issued by the Oracle server.

To configure the Oracle monitor:

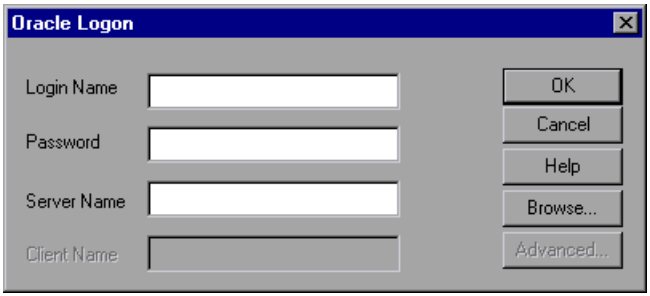
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

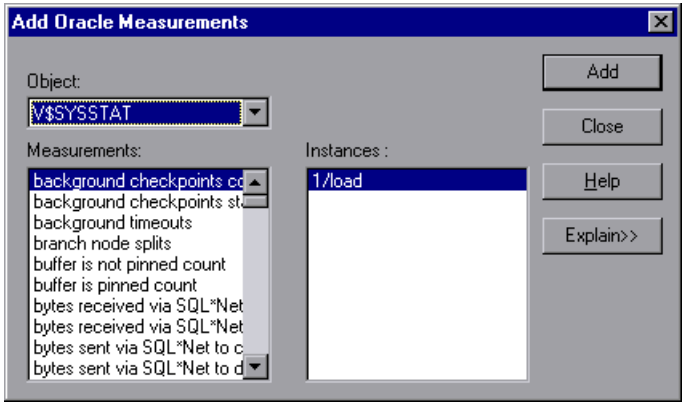
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Database Server Resource category, select **Oracle**, and then click **Add**.

The Oracle Logon dialog box opens.



5 Enter your Login Name, Password, and Server Name, and click **OK**. The Add Oracle Measurements dialog box opens.



6 Select an object, a measurement, and an instance. You can select multiple measurements using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted measurement are running. For a description of each measurement, click **Explain>>** to expand the dialog box. For instructions on creating custom queries, see "Custom Queries," on page 508.

For a list of the available performance counters, see **page 507**.

7 Click **Add** to place the selected measurement on the resource list. Add all the desired resources to the list, and click **Close**.

8 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: By default, the database returns the absolute value of a counter. However, by changing the IsRate setting in the *dat\monitors\vmon.cfg* file to 1, you can instruct the database to report a counter's rate value—the change in the counter per unit time.

Oracle Performance Counters

The following measurements are most commonly used when monitoring the Oracle server (from the V$SYSSTAT table):

Measurement	Description
CPU used by this session	This is the amount of CPU time (in 10s of milliseconds) used by a session between the time a user call started and ended. Some user calls can be completed within 10 milliseconds and, as a result, the start and end user-call time can be the same. In this case, 0 milliseconds are added to the statistic. A similar problem can exist in the operating system reporting, especially on systems that suffer from many context switches.
Bytes received via SQL*Net from client	The total number of bytes received from the client over Net8
Logons current	The total number of current logons
Opens of replaced files	The total number of files that needed to be reopened because they were no longer in the process file cache

Measurement	Description
User calls	Oracle allocates resources (Call State Objects) to keep track of relevant user call data structures every time you log in, parse, or execute. When determining activity, the ratio of user calls to RPI calls gives you an indication of how much internal work gets generated as a result of the type of requests the user is sending to Oracle.
SQL*Net roundtrips to/from client	The total number of Net8 messages sent to, and received from, the client
Bytes sent via SQL*Net to client	The total number of bytes sent to the client from the foreground process(es)
Opened cursors current	The total number of current open cursors
DB block changes	Closely related to consistent changes, this statistic counts the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These are changes that are generating redo log entries and hence will be permanent changes to the database if the transaction is committed. This statistic is a rough indication of total database work and indicates (possibly on a per-transaction level) the rate at which buffers are being dirtied.
Total file opens	The total number of file opens being performed by the instance. Each process needs a number of files (control file, log file, database file) to work against the database.

Custom Queries

Using the custom query feature, you can define your own query to the Oracle database and view the result of this query—a single numerical value—in the Oracle online monitor graph. By defining your own query, you can monitor not only the V$SYSSTAT and V$SESSTAT table counters that are currently provided by the Oracle monitor, but other tables that contain useful performance information as well.

To create a custom query:

1. In the third line of the *vmon.cfg* file, CustomCounters=, indicate the number of custom counters you want to create.
2. Create a new section in the *vmon.cfg* file for the new counter. Each section has the following format:

   ```
   [Custom2]
   Name=Number of sessions
   Description=This counter returns the number of sessions active.
   Query=SELECT COUNT(*) FROM V$SESSION
   IsRate=1
   ```

3. In the [Custom#] line, assign the next number in the sequence of counters to the new custom counter. Note that the custom counters must be in consecutive order, beginning with the number 0.
4. In the Name line, enter the name of the new counter.
5. In the Description line, enter the description of the counter that you want the help message to contain.
6. In the Query line, enter the text of the SQL query (on one line of the *vmon.cfg* file) that returns exactly one row from the database. This row must contain one column, a numerical value.

Note: Custom queries should not exceed 512 characters.

---

7. In the IsRate line, enter 0 if you want the database to report the counter as an absolute number. If you want the database to report the change in the counter per unit time, enter 1.

Note: Custom queries cannot return negative values.

Configuring the SQL Server Monitor

The SQL Server monitor measures the standard Windows resources on the SQL server machine.

Note: To monitor an SQL server through a firewall, use TCP, port 139.

To configure the SQL server monitor:

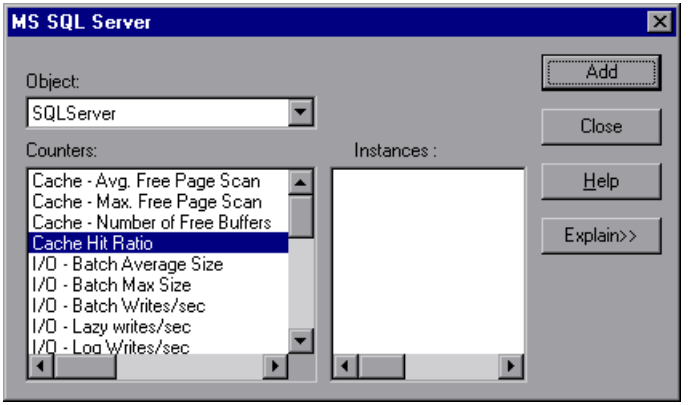
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Database Server Resource category, select **MS SQL Server**, and then click **Add**.

A dialog box displaying the SQL Server object, its counters, and instances opens.



For a list of the available performance counters, see **page 512**.

Note: To change the default counters for the SQL Server monitor, see "Changing a Monitor's Default Counters," on page 775.

---

5 Select a counter and an instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: Certain measurements or counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on the SQL Server. For more information about these counters, see "Useful Counters for Stress Testing," on page 776.

SQL Server Performance Counters

The following table describes the default counters that can be monitored on version 6.5 of the SQL Server:

Measurement	Description
% Total Processor Time (NT)	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4th of the processors are 100% busy this is 25%. It can be viewed as the fraction of the time spent doing useful work. Each processor is assigned an Idle thread in the Idle process which consumes those unproductive processor cycles not used by any other threads.
% Processor Time (Win 2000)	The percentage of time that the processor is executing a non-idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting that value from 100%. (Each processor has an idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.
Cache Hit Ratio	The percentage of time that a requested data page was found in the data cache (instead of being read from disk)
I/O - Batch Writes/sec	The number of 2K pages written to disk per second, using Batch I/O. The checkpoint thread is the primary user of Batch I/O.
I/O - Lazy Writes/sec	The number of 2K pages flushed to disk per second by the Lazy Writer
I/O - Outstanding Reads	The number of physical reads pending

Measurement	Description
I/O - Outstanding Writes	The number of physical writes pending
I/O - Page Reads/sec	The number of physical page reads per second
I/O - Transactions/sec	The number of Transact-SQL command batches executed per second
User Connections	The number of open user connections

Configuring the Sybase Monitor

The Sybase monitor enables monitoring of Sybase Adaptive Server Enterprise (Sybase ASE) servers (version 11 or later) on Windows and UNIX. The monitor connects to the Sybase ASE server (via the Adaptive Server Enterprise Monitor Server) and retrieves metrics from the server using standard, Sybase-provided libraries.

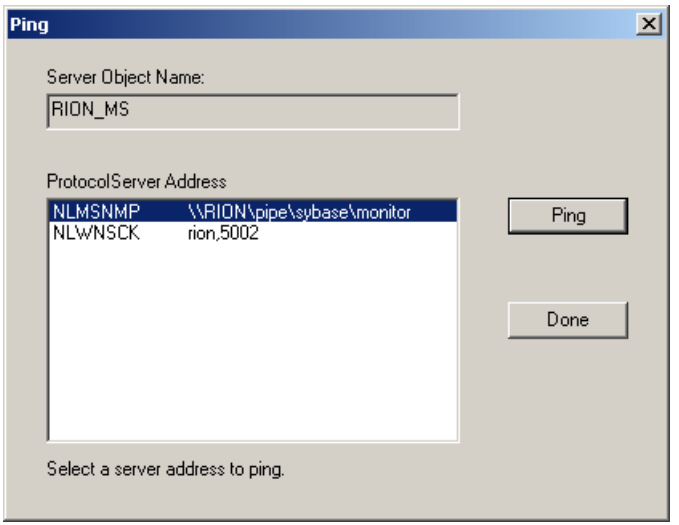
Note: When connecting to the monitored server, you connect to the Adaptive Server Enterprise Monitor Server, not the Sybase ASE server. The Adaptive Server Enterprise Monitor Server is an application that runs on the same machine as Sybase ASE server and retrieves performance information from it. The Adaptive Server Enterprise Monitor Server usually has the same name as the Sybase server, but with the suffix **_ms**.

In order to monitor the Sybase ASE server, you must first set up the Sybase monitor environment.

To set up the Sybase monitor environment:

1 Install the Sybase client files and libraries on the Console machine.

2 Verify a connection between the client and server on the Console machine. To do so, use the Sybase client's *dsedit* tool to ping the Adaptive Server Enterprise Monitor Server.



Note: The port you use to monitor a Sybase server through a firewall depends on the configuration of the Sybase server. Configuration information for the connection between the client and server is located in the Sybase client *sql.ini* file.

To configure the Sybase ASE monitor:

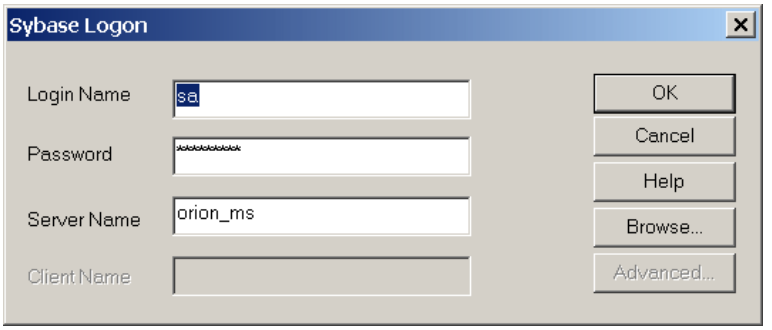
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

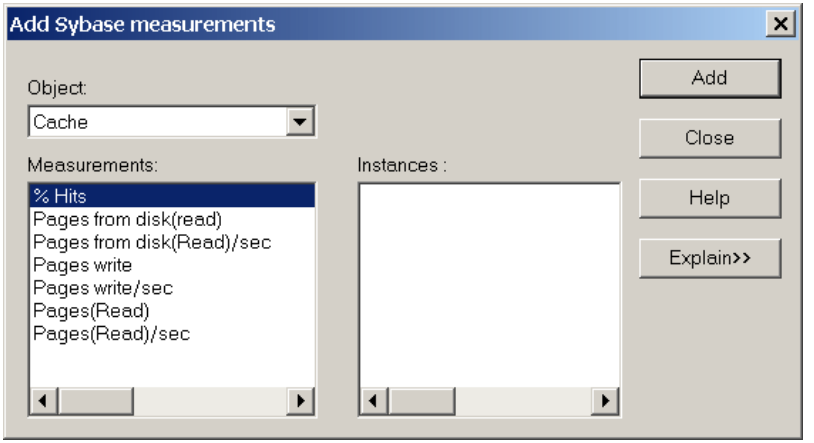
4 In the left section of the dialog box, expand the Database Server Resource category, select **Sybase**, and then click **Add**.

The Sybase Logon dialog box opens.



5 Enter the login name and password of a user that has administrative privileges on the Sybase ASE server, as well as the Adaptive Server Enterprise Monitor Server name (usually the same name as the Sybase server but with the suffix *_ms*).

6 Click **OK**. The Add Sybase Measurements dialog box opens.



7 Select an object, measurement, and instance. You can select multiple measurements using the **CTRL** key. The instance is relevant only if multiple instances of the highlighted measurement are running. For a description of the measurements, click **Explain>>** to expand the dialog box.

For a list of the available performance counters, see **page 516**.

8 Click **Add** to place the selected measurement on the resource list. Add all the desired resources to the list, and click **Close**.

9 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Sybase Performance Counters

The following measurements are available when monitoring a Sybase server:

Object	Measurement	Description
Network	Average packet size (Read)	Reports the number of network packets received
	Average packet size (Send)	Reports the number of network packets sent
	Network bytes (Read)	Reports the number of bytes received, over the sampling interval
	Network bytes (Read)/sec	Reports the number of bytes received, per second
	Network bytes (Send)	Reports the number of bytes sent, over the sampling interval
	Network bytes (Send)/sec	Reports the number of bytes sent, per second
	Network packets (Read)	Reports the number of network packets received, over the sampling interval
	Network packets (Read)/sec	Reports the number of network packets received, per second
	Network packets (Send)	Reports the number of network packets sent, over the sampling interval
	Network packets (Send)/sec	Reports the number of network packets sent, per second
Memory	Memory	Reports the amount of memory, in bytes, allocated for the page cache

Object	Measurement	Description
Disk	Reads	Reports the number of reads made from a database device
	Writes	Reports the number of writes made to a database device
	Waits	Reports the number of times that access to a device had to wait
Disk	Grants	Reports the number of times access to a device was granted
Engine	Server is busy (%)	Reports the percentage of time during which the Adaptive Server is in a "busy" state
	CPU time	Reports how much "busy" time was used by the engine
	Logical pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device
	Pages from disk (Read)	Reports the number of data page reads that could not be satisfied from the data cache
	Pages stored	Reports the number of data pages written to a database device
Stored Procedures	Executed (sampling period)	Reports the number of times a stored procedure was executed, over the sampling interval
	Executed (session)	Reports the number of times a stored procedure was executed, during the session
	Average duration (sampling period)	Reports the time, in seconds, spent executing a stored procedure, over the sampling interval
	Average duration (session)	Reports the time, in seconds, spent executing a stored procedure, during the session

Object	Measurement	Description
Locks	% Requests	Reports the percentage of successful requests for locks
	Locks count	Reports the number of locks. This is an accumulated value.
Locks	Granted immediately	Reports the number of locks that were granted immediately, without having to wait for another lock to be released
	Granted after wait	Reports the number of locks that were granted after waiting for another lock to be released
	Not granted	Reports the number of locks that were requested but not granted
	Wait time (avg.)	Reports the average wait time for a lock
SqlSrvr	Locks/sec	Reports the number of locks. This is an accumulated value.
	% Processor time (server)	Reports the percentage of time that the Adaptive Server is in a "busy" state
	Transactions	Reports the number of committed Transact-SQL statement blocks (transactions)
	Deadlocks	Reports the number of deadlocks

Object	Measurement	Description
Cache	% Hits	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read
	Pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device
	Pages (Read)/sec	Reports the number of data page reads, whether satisfied from cache or from a database device, per second
Cache	Pages from disk (Read)	Reports the number of data page reads that could not be satisfied from the data cache
	Pages from disk (Read)/sec	Reports the number of data page reads, per second, that could not be satisfied from the data cache
	Pages (Write)	Reports the number of data pages written to a database device
	Pages (Write)/sec	Reports the number of data pages written to a database device, per second

Object	Measurement	Description
Process	% Processor time (process)	Reports the percentage of time that a process running a given application was in the "Running" state (out of the time that all processes were in the "Running" state)
	Locks/sec	Reports the number of locks, by process. This is an accumulated value.
	% Cache hit	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read, by process
	Pages (Write)	Reports the number of data pages written to a database device, by process
Transaction	Transactions	Reports the number of committed Transact-SQL statement blocks (transactions), during the session
	Rows (Deleted)	Reports the number of rows deleted from database tables during the session

Object	Measurement	Description
Transaction	Inserts	Reports the number of insertions into a database table during the session
	Updates	Reports the updates to database tables during the session
	Updates in place	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred) during the session
	Transactions/sec	Reports the number of committed Transact-SQL statement blocks (transactions) per second
	Rows (Deleted)/sec	Reports the number of rows deleted from database tables, per second
	Inserts/sec	Reports the number of insertions into a database table, per second
	Updates/sec	Reports the updates to database tables, per second
	Updates in place/sec	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred), per second

35

Streaming Media Monitoring

During a session run, you can monitor the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer and Media Player clients, in order to isolate server and client performance bottlenecks.

This chapter describes:

- Configuring the Windows Media Server Monitor
- Configuring the RealPlayer Server Monitor
- Viewing the RealPlayer Client Online Graph
- Viewing the Media Player Client Online Graph

Note: For instructions on recording a script containing streaming media functions, refer to the *Creating Vuser Scripts* guide.

About Streaming Media Monitoring

The streaming media monitors provide you with performance information for the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer and Media Player clients. In order to obtain data for the Windows Media Server and RealPlayer Server, you need to activate the streaming media monitor before executing the session, and indicate which statistics and measurements you want to monitor. The RealPlayer Client and Media Player Client do not require pre-session activation or configuration.

Configuring the Windows Media Server Monitor

To monitor the Windows Media Server, you must first select the counters you want the Windows Media Server monitor to measure. You select these counters using the Windows Media Server dialog box.

To configure the Windows Media Server monitor:

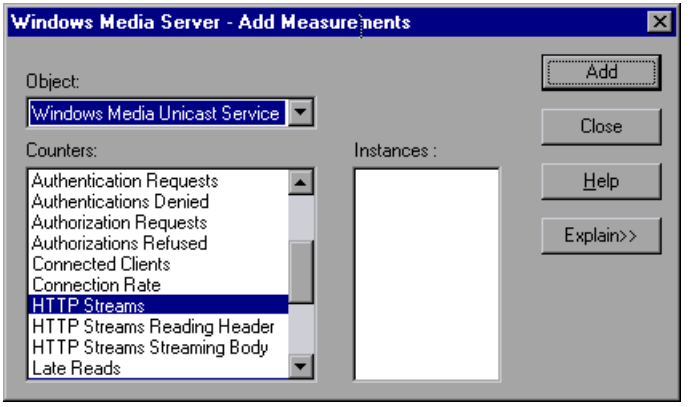
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Streaming Media Server category, select **Windows Media Server**, and then click **Add**.

The Windows Media Server dialog box opens, displaying the Windows Media Unicast Service object, its counters, and instances.



The following table describes the default counters that can be monitored:

Measurement	Description
Active Live Unicast Streams (Windows)	The number of live unicast streams that are being streamed
Active Streams	The number of streams that are being streamed
Active TCP Streams	The number of TCP streams that are being streamed
Active UDP Streams	The number of UDP streams that are being streamed
Aggregate Read Rate	The total, aggregate rate (bytes/sec) of file reads
Aggregate Send Rate	The total, aggregate rate (bytes/sec) of stream transmission
Connected Clients	The number of clients connected to the server
Connection Rate	The rate at which clients are connecting to the server
Consoles	The number of Consoles currently connected to the server
HTTP Streams	The number of HTTP streams being streamed
Late Reads	The number of late read completions per second
Pending Connections	The number of clients that are attempting to connect to the server, but are not yet connected. This number may be high if the server is running near maximum capacity and cannot process a large number of connection requests in a timely manner.
Stations	The number of station objects that currently exist on the server
Streams	The number of stream objects that currently exist on the server
Stream Errors	The cumulative number of errors occurring per second

5 Select a counter and an instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

For a list of the available performance counters, see page 526.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Windows Media Server Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Active Live Unicast Streams (Windows)	The number of live unicast streams that are being streamed
Active Streams	The number of streams that are being streamed
Active TCP Streams	The number of TCP streams that are being streamed
Active UDP Streams	The number of UDP streams that are being streamed
Aggregate Read Rate	The total, aggregate rate (bytes/sec) of file reads
Aggregate Send Rate	The total, aggregate rate (bytes/sec) of stream transmission
Connected Clients	The number of clients connected to the server
Connection Rate	The rate at which clients are connecting to the server
Consoles	The number of Consoles currently connected to the server
HTTP Streams	The number of HTTP streams being streamed
Late Reads	The number of late read completions per second
Pending Connections	The number of clients that are attempting to connect to the server, but are not yet connected. This number may be high if the server is running near maximum capacity and cannot process a large number of connection requests in a timely manner.
Stations	The number of station objects that currently exist on the server

Measurement	Description
Streams	The number of stream objects that currently exist on the server
Stream Errors	The cumulative number of errors occurring per second

Configuring the RealPlayer Server Monitor

To monitor the RealPlayer Server, you must first select the counters you want the RealPlayer Server monitor to measure. You select these counters using the Real Server dialog box.

To configure the RealPlayer Server monitor:

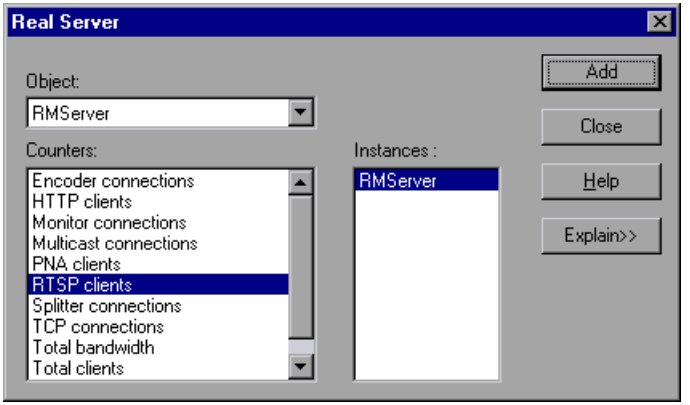
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Streaming Media Server category, select **RealPlayer Server**, and then click **Add**.

The Real Server dialog box opens, displaying the counters that can be monitored.



5 Select a counter and an instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted counter are running. For a description of each counter, click **Explain**>> to expand the dialog box.

For a list of the available performance counters, see page 529.

6 Click **Add** to place the selected counter on the resource list. Add all the desired resources to the list, and click **Close**.

7 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Real Server Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Encoder Connections	The number of active encoder connections
HTTP Clients	The number of active clients using HTTP
Monitor Connections	The number of active server monitor connections
Multicast Connections	The number of active multicast connections
PNA Clients	The number of active clients using PNA
RTSP Clients	The number of active clients using RTSP
Splitter Connections	The number of active splitter connections
TCP Connections	The number of active TCP connections
Total Bandwidth	The number of bits per second being consumed
Total Clients	The total number of active clients
UDP Clients	The number of active UDP connections

Viewing the RealPlayer Client Online Graph

To view the RealPlayer Client online monitor graph:

1. Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.
2. Click the "+" next to **Streaming Media** to expand the graph tree, and select the **Real Client** graph.
3. Click **Open Graph**. The graph appears in the graph view area.

The following table describes the RealPlayer Client measurements that are monitored:

Measurement	Description
Current Bandwidth (Kbits/sec)	The number of kilobytes in the last second
Buffering Event Time (sec)	The average time spent on buffering
Network Performance	The ratio (percentage) between the current bandwidth and the actual bandwidth of the clip
Percentage of Recovered Packets	The percentage of error packets that were recovered
Percentage of Lost Packets	The percentage of packets that were lost
Percentage of Late Packets	The percentage of late packets
Time to First Frame Appearance (sec)	The time for first frame appearance (measured from the start of the replay)
Number of Buffering Events	The average number of all buffering events
Number of Buffering Seek Events	The average number of buffering events resulting from a seek operation
Buffering Seek Time	The average time spent on buffering events resulting from a seek operation

Measurement	Description
Number of Buffering Congestion Events	The average number of buffering events resulting from network congestion
Buffering Congestion Time	The average time spent on buffering events resulting from network congestion
Number of Buffering Live Pause Events	The average number of buffering events resulting from live pause
Buffering Live Pause Time	The average time spent on buffering events resulting from live pause

Viewing the Media Player Client Online Graph

To view the Media Player Client online monitor graph:

1. Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.
2. Click the "+" next to **Streaming Media** to expand the graph tree, and select the **Media Player Client** graph.
3. Click **Open Graph**. The graph appears in the graph view area.

The following table describes the Media Player Client measurements that are monitored:

Measurement	Description
Average Buffering Events	The number of times Media Player Client had to buffer incoming media data due to insufficient media content
Average Buffering Time (sec)	The time spent by Media Player Client waiting for sufficient amount of media data in order to continue playing media clip
Current bandwidth (Kbits/sec)	The number of kbits per second received

Measurement	Description
Number of Packets	The number of packets sent by server for a particular media clip
Stream Interruptions	The number of interruptions encountered by media player client while playing a media clip. This measurement includes the number of times Media Player Client had to buffer incoming media data, and any errors that occurred during playback.
Stream Quality (Packet-level)	The percentage ratio of packets received to total packets
Stream Quality (Sampling-level)	The percentage of stream samples received on time (no delays in reception)
Total number of recovered packets	The number of lost packets that were recovered. This value is only relevant during network playback.
Total number of lost packets	The number of lost packets that were not recovered. This value is only relevant during network playback.

36

ERP/CRM Server Resource Monitoring

During a session step run, you can monitor ERP server resources in order to isolate server performance bottlenecks.

This chapter explains how to configure the SAP, SAP Portal, Siebel, and Siebel Server Manager ERP/CRM Server Resource Monitors.

It describes:

- Setting up the Monitoring Environment
- Configuring the SAP Monitor
- Configuring the SAP Portal Monitor
- Configuring the Siebel Web Server Monitor
- Configuring the Siebel Server Manager Monitor
- Configuring the PeopleSoft (Tuxedo) Monitor
- PeopleSoft Ping Monitoring

About ERP/CRM Server Resource Monitoring

Siebel, Siebel Server Manager, SAP Portal, and PeopleSoft monitors are ERP/CRM Server Resource Monitors, that provide performance information such as the number of open sessions, active transactions, and database connections for the Siebel, SAP Portal and PeopleSoft ERP/CRM application servers. You can monitor the server resources on a machine during a session step run, and determine why a bottleneck occurred on a particular machine.

Setting up the Monitoring Environment

To monitor the Siebel, Siebel Server Manager, SAP, SAP Portal and PeopleSoft server performance, you must first install and configure SiteScope. SiteScope is the application that is used to monitor these servers.

Before setting up the ERP/CRM Server Resource monitors:

1 Make sure that SiteScope has been installed on a dedicated machine.

Note: It is recommended that you install SiteScope and the LoadRunner Tuning Module Console on different machines.

---

2 On the machine where SiteScope is installed, configure SiteScope to monitor the required ERP/CRM server machines. For more information on configuring the SiteScope server, refer to the SiteScope User Guide (http://www.freshwater.com/SiteScope/UGtoc.htm)or see the appropriate section:

- ➤ "Configuring the SAP Portal Monitor on the SiteScope Machine" on page 541.
- ➤ "Configuring the Siebel Web Server Monitor on the SiteScope Machine" on page 544.
- ➤ "Configuring the Siebel Server Manager Client on the SiteScope Machine" on page 548.

Note: SiteScope's default sampling rate is 10 minutes, and its minimum rate 15 seconds.

---

3 Verify that SiteScope is collecting the required data from the servers it is monitoring.

4 In the System Topology window, add an element representing the server running Siebel. Note that this element must not be a Load Generator.

Configuring the SAP Monitor

In order to obtain data on the SAP R/3 system server , you need to activate the ERP/CRM server resource monitor before executing the session step, and indicate which statistics and measurements you want to monitor.

You can use the SAP monitor to view:

- the number of configured instances for each SAP system
- data for all application instances (not just the one you logged on to)
- transactions used and the users that call them
- number of users working on the different instances
- performance history for recent periods of all instances
- response time distribution
- resource consumption for any application server
- application server workload for today or for a recent period.

Before you can monitor the SAP R/3 system server, you must set up the SAP monitor environment.

Setting Up the SAP Monitor

Before monitoring a SAP R/3 system server, you must set up the server monitor environment.

To set up the SAP monitor environment:

1. Install the SAP GUI client on the Console machine.
2. Click **F6** to check whether you can access the st03 transaction and query for *last minute load* information. If this functionality is not already enabled, enable it from the SAP R/3 client on the Console machine, using the username and password defined in the Console.

Configuring the SAP Monitor

To monitor a SAP R/3 system server, you must select the counters you want the SAP monitor to measure. You select these counters using the Add SAP Monitor Measurements dialog box.

Note: The SAP R/3 performance monitor supports SAP server versions 3.1 to 4.6, regardless of the SAP R/3 server's operating system and the platform on which it is installed.

To configure the SAP monitor on the Console machine:

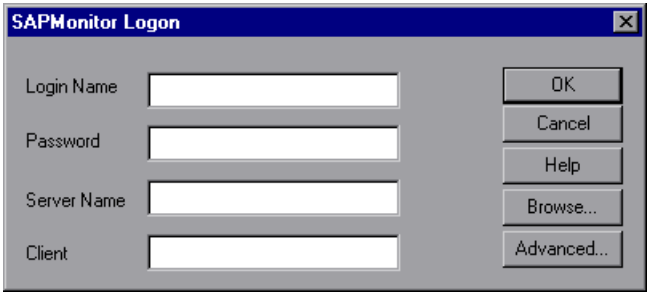
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running SAP.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **SAP**, and then click **Add**.

The SAP Monitor Logon dialog box opens.



5 Enter your Login Name, Password, Server Name, and Client.

Note: If you want to connect to the SAP monitor through a router, you need to enter the router string into the Server Name field. A router string has the format:
<RouterString/ServerIP/S/sapdpxx>

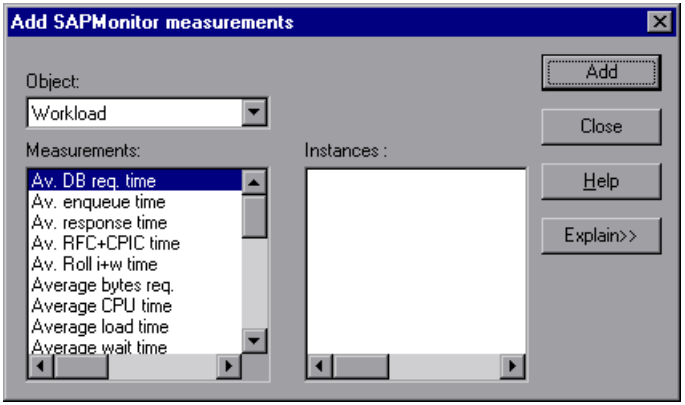
where RouterString is /H/<IP_ADDRESS>/H/<IP_ADDRESS>/H/
ServerIP is the application server IP address
and *xx* is the system number.

For example, if the router string = /H/199.35.107.9/H/204.79.199.244/H/, application server IP address = 172.20.11.6, and the system number = 00, you should enter the following string into the Server Name field:

/H/199.35.107.9/H/204.79.199.244/H/172.20.11.6/S/sapdp00

---

6 Click **OK**. The Add SAP Monitor Measurements dialog box opens.



7 Select an object, a measurement, and an instance. You can select multiple measurements using the **Ctrl** key. The instance is relevant only if multiple instances of the highlighted measurement are running. For a description of each measurement, click **Explain>>** to expand the dialog box.

For a list of the available performance counters, see page 538.

8 Click **Add** to place the selected measurement on the resource list. Add all the desired resources to the list, and click **Close**.

9 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

SAP Performance Counters

The following are the most commonly monitored counters:

Measurement	Description
Average CPU time	The average CPU time used in the work process.
Average response time	The average response time, measured from the time a dialog sends a request to the dispatcher work process, through the processing of the dialog, until the dialog is completed and the data is passed to the presentation layer. The response time between the SAP GUI and the dispatcher is not included in this value.
Average wait time	The average amount of time that an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects, such as ABAP source code and screen information, from the database.
Database calls	The number of parsed requests sent to the database.

Measurement	Description
Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual database calls. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of calls/requests gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.
Roll ins	The number of rolled-in user contexts.
Roll outs	The number of rolled-out user contexts.
Roll in time	The processing time for roll ins.
Roll out time	The processing time for roll outs.
Roll wait time	The queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them.
Average time per logical DB call	The average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, the network, the buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

Configuring the SAP Portal Monitor

To use the SAP Portal monitor, you must first configure the SAP Portal monitor on the SiteScope machine, and then select the counters you want the SAP Portal monitor to measure. You select these counters using the Console's SAP Portal dialog box.

Before monitoring a SAP Portal server, you must set up the server monitor environment.

Before setting up the SAP Portal monitor:

1. Make sure that SiteScope has been installed on a server. SiteScope is the application that is used to monitor the SAP Portal server. Although you can install SiteScope on the Console machine, we recommend installing it on a dedicated server.
2. On the machine where SiteScope is installed, configure SiteScope to monitor the required SAP Portal machines. For more information on configuring the SiteScope server, see "Configuring the SAP Portal Monitor on the SiteScope Machine" on page 541.

Note: When you assign a name to a monitor, include the server name in the monitor name. This avoids any confusion as to which host the monitor belongs.

Note: When you define the name of the monitor on the SiteScope server, do not use "\" in the Title field (e.g., CPU Utilization on \\superman).

---

3. Verify that SiteScope is collecting the required data from the servers it is monitoring. From the SiteScope Panel, select the monitor group polling the SAP Portal server machines, and check that the monitor displays a list of server measurements in the Status column.

To configure the SAP Portal monitor on the SiteScope machine:

1. Restart SiteScope after installing the monitor add-in, and verify connectivity to the SAP Portal SWSE page by opening the following URL from the machine where sitescope is installed:

   http://<*your_SAP_Portal_server*>/sapportal

2. In the SiteScope Panel, click **Create Group**, enter a name for the group, and click **Add**.
3. In the **Add to Group** section, click **Monitor** and select **SAP Portal Monitor** from the list of monitors.
4. In the server field, enter the URL to the SWE stats page you want to monitor. For example:

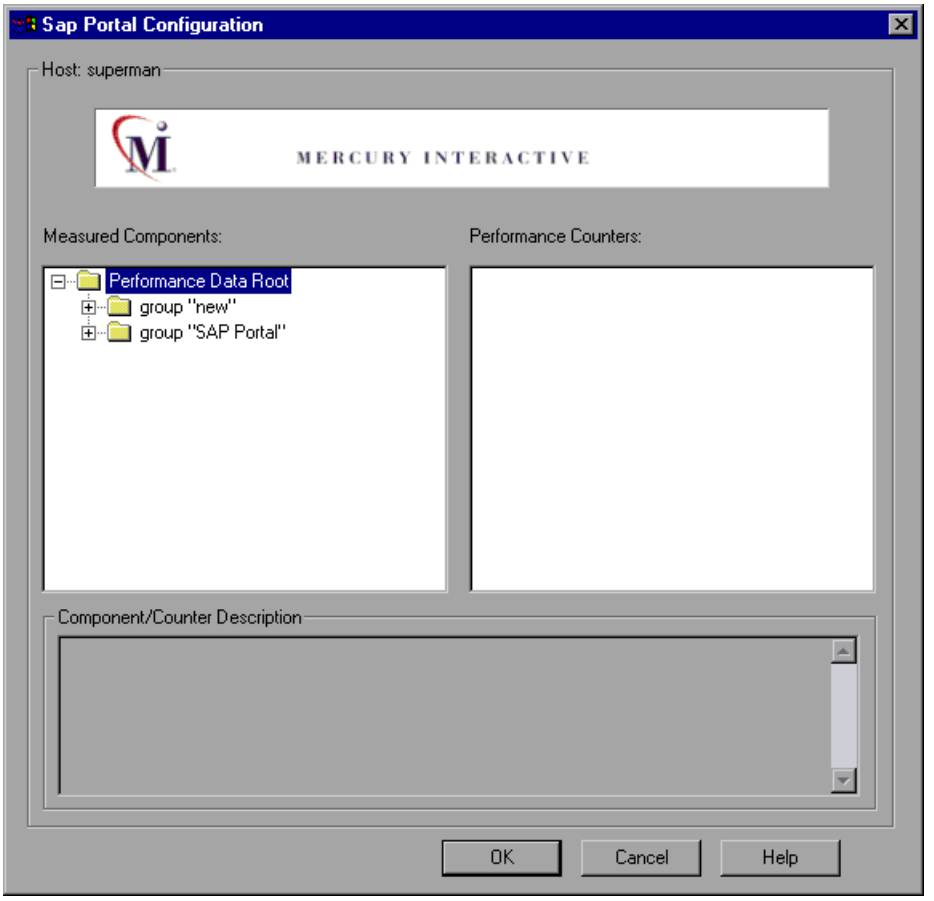
   http://<*your_SAP_Portal_server*>/sapportal

5. Enter the name of the application you want to monitor in the Application field.
6. Enter the user name and password (if applicable).
7. Click **Choose Counters** and select your desired counters.
8. Click **Choose Counters** again, and then click **Add Monitor**.

To configure the SAP Portal monitor on the Console machine:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the ERP-CRM Server category, select **SAP Portal**, and then click **Add**.

The SAP Portal Monitor Configuration dialog box is displayed.



5 In the Measured Components pane, locate the SAP Portal measurement that you are monitoring and click it. The performance counters that SAP Portal is monitoring on the selected component are displayed in the Performance Counters pane.

For a list of measurements, see "SAP Portal Performance Counters," on page 543.

6 Check the required performance counters in the Performance Counters pane. When you have selected the performance counters for the SAP Portal measurements you are monitoring, click **OK** to close the SAP Portal

Configuration dialog box. The **Select Measurements to Monitor** dialog box appears with the selected SAP Portal measurements in the **Selected Measurements** pane.

7 Click **OK** in the **Select Measurements to Monitor** dialog box, and click **OK** in the Monitors Configuration dialog box, to activate the SAP Portal monitor.

Note: For troubleshooting tips and limitations, see "Troubleshooting Server Resource Monitors," on page 645.

SAP Portal Performance Counters

The following table shows the default counters that can be measured:

Measurement	Description
Accumulated Amount of Outbound Data (bytes)	The accumulated amount of outbound data, measured in bytes.
Time for all Requests (ms)	The total time, in milliseconds, taken for processing all requests.
Average Amount of Outbound Data per Request (bytes)	The average amount of outbound data per request, measured in bytes.
Average Number of Component Calls per Request (bytes)	The average number of component calls per request, measured in bytes.
Average Time of a Request (ms)	The average amount of time, in milliseconds, taken to process a request.
Number of Calls with Outbound Data	The total number of calls with outbound data.
Number of Component Calls for all Requests	The total number of component calls for all requests.
Number of Requests since First Request	The total number of requests since the first request was made.

Measurement	Description
Requests per Second	The number of requests made per second.
Time Stamp of First Request	The time stamp of the first request.

Configuring the Siebel Web Server Monitor

To use the Siebel monitor, you must first configure the Siebel monitor on the SiteScope machine, and then select the counters you want the Siebel monitor to measure. You select these counters using the Console's Siebel dialog box.

To configure the Siebel Web Server monitor on the SiteScope machine:

1. Restart SiteScope after installing the monitor add-in, and verify connectivity to the Siebel SWSE page by opening the following URL from the machine where sitescope is installed:

   http://<*your_siebel_server*>/callcenter/_stats.swe

2. In the SiteScope Panel, click **Create Group**, enter a name for the group, and click **Add**.
3. In the **Add to Group** section, click **Monitor** and select **Siebel Web Server** from the list of monitors.
4. In the server field, enter the URL to the SWE stats page you want to monitor. For example:

   http://<*your_siebel_server*>/callcenter/_stats.swe

5. Enter the name of the application you want to monitor in the Application field. For example: Callcenter.
6. Enter the user name and password (if applicable).
7. Click **Choose Counters** and select your desired counters.
8. Click **Choose Counters** again, and then click **Add Monitor**.

To configure the Siebel Web Server monitor on the Console machine:

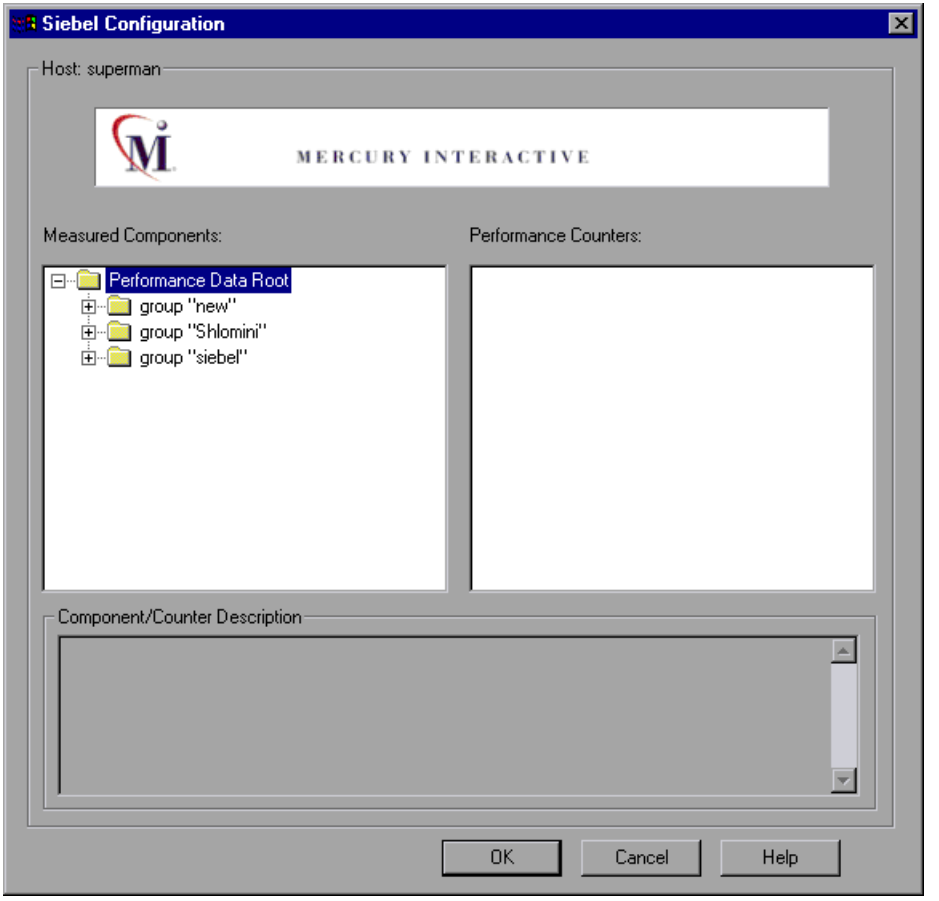
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running Siebel.

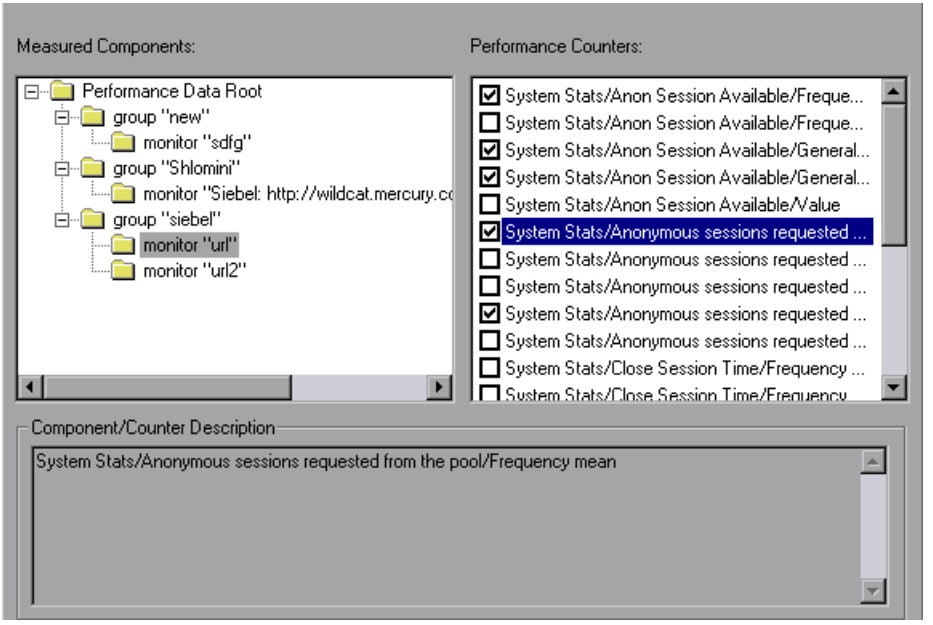
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **Siebel Web Server**, and then click **Add**.

The Siebel Monitor Configuration dialog box is displayed.



5 In the Measured Components pane, locate the Siebel measurement that you are monitoring and click it. The performance counters that Siebel is monitoring on the selected component are displayed in the Performance Counters pane.



For a list of the available performance counters, see page 547.

6 Check the required performance counters in the Siebel Monitor Configuration window's right pane.

7 When you have selected the performance counters for the Siebel measurements you are monitoring, click **OK** to close the Siebel Monitor Configuration dialog box. The **Select Measurements to Monitor** dialog box appears with the selected Siebel measurements in the **Selected Measurements** pane.

8 Click **OK** in the **Select Measurements to Monitor** dialog box, and click **OK** in the Monitors Configuration dialog box, to activate the Siebel monitor.

Note: For troubleshooting tips and limitations, see "Troubleshooting Server Resource Monitors," on page 645.

Siebel Web Server Performance Counters

The following table shows the default counters that can be measured:

Measurement	Description
Anonymous sessions requested from the pool	The number of anonymous sessions requested from the pool.
Open Session Time	The time users experience logging on to the system.
Anon Session Removed	The number of anonymous sessions removed from the pool.
Anon Session Available	The number of anonymous sessions available in the pool.
Anonymous sessions returns to the pool	The number of anonymous sessions returned to the pool.
Response Time	The time taken to respond to a user request.
Close Session Time	The time users experience logging off the system.
Request Time	The time taken to process the user request.

Configuring the Siebel Server Manager Monitor

To monitor the Siebel Server Manager performance, you must first install the Siebel Server Manager client on the SiteScope machine. Then select the counters you want the Siebel Server Manager monitor to measure. You select these counters using the Console's Siebel Server Manager dialog box.

To configure the Siebel Server Manager client on the SiteScope machine:

1. Restart SiteScope and verify connectivity to the Siebel SWE page by opening the following URL from the machine where SiteScope is installed:

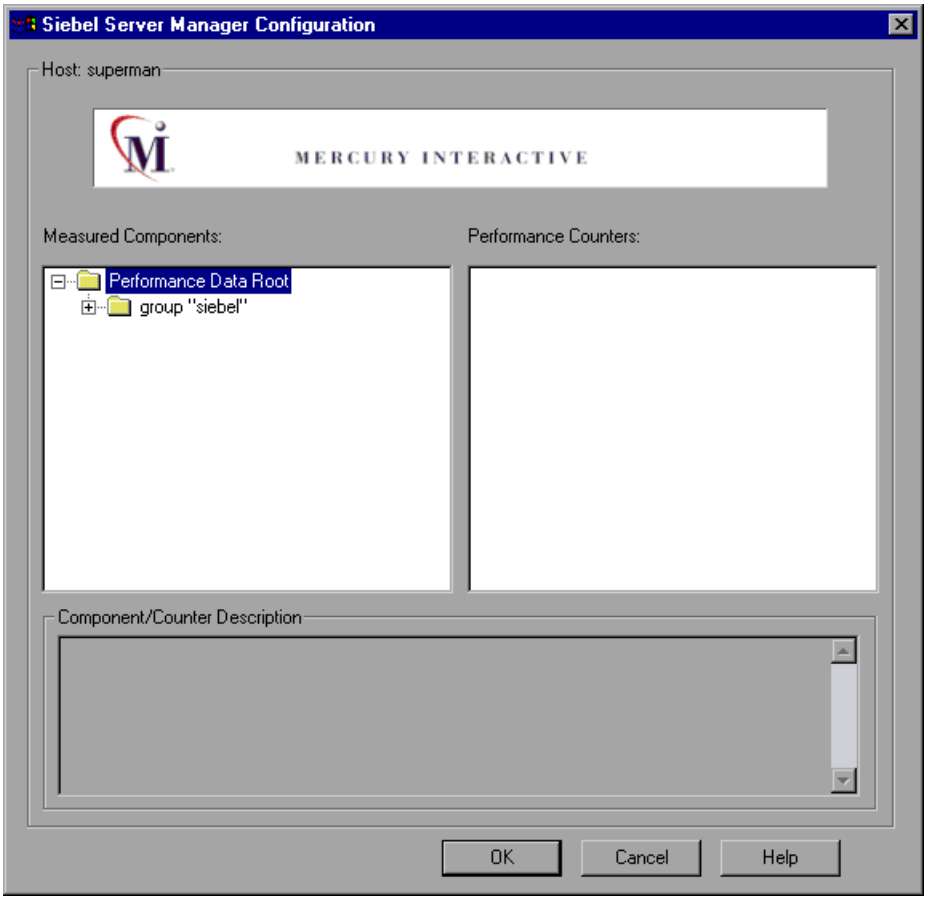
   http://<*your_siebel_server*>/<*application_name*>/_stats.swe

2. In the SiteScope Panel, click **Create Group**, enter a name for the group, and click **Add**.
3. In the **Add to Group** section, click **Monitor** and select **Siebel Server Manager** from the list of monitors.
4. Enter the name of the Siebel Server in the Application server field, the Enterprise Server in the Enterprise server field, and the Gateway Server in the Gateway server field.
5. Enter the path to Siebel Server Manager in the Path to Script field.
6. Enter the user name and password (if applicable).
7. Click **Choose Counters** and select your desired counters.
8. Click **Choose Counters** again, and then click **Add Monitor**.

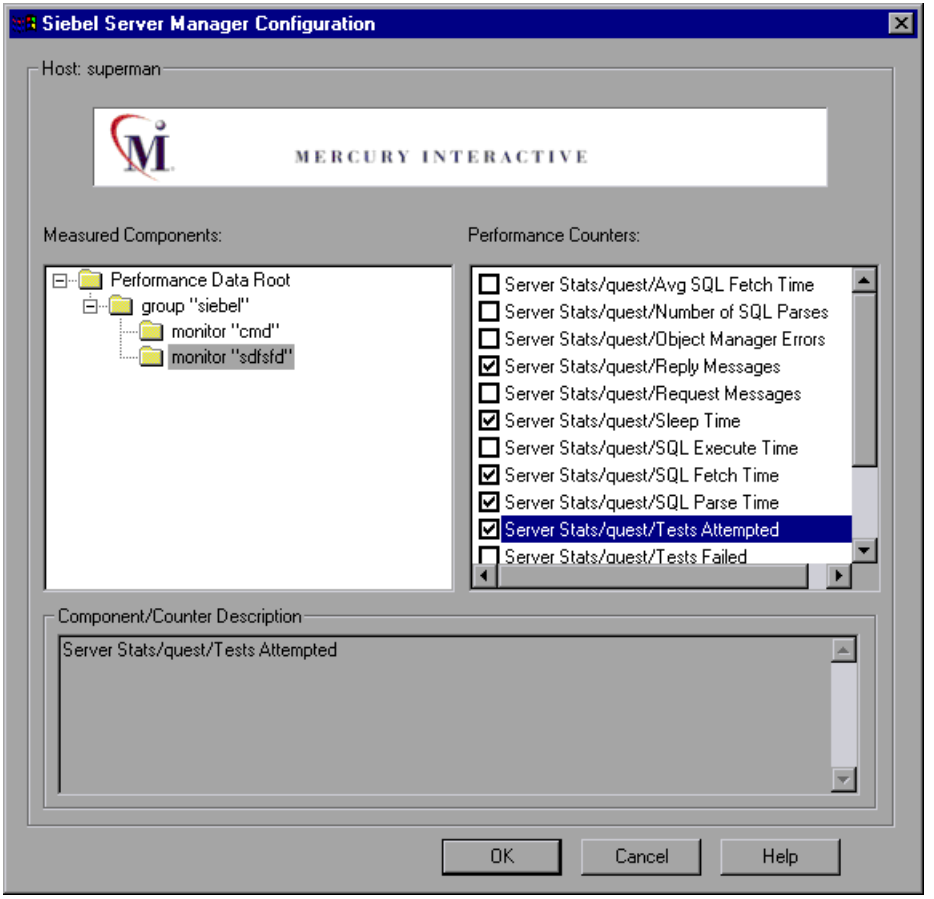
To configure the Siebel Server Manager monitor on the Console machine:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the server running Siebel.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the ERP-CRM Server category, select **Siebel Server Manager**, and then click **Add**.

The Siebel Server Manager Monitor Configuration dialog box is displayed.



5 In the Measured Components pane, locate the Siebel Server Manager measurement that you are monitoring and click it. The performance counters that Siebel Server Manager is monitoring on the selected component are displayed in the Performance Counters pane.



6 Check the required performance counters in the Siebel Server Manager Monitor Configuration window's right pane.

For a list of the available performance counters, see page 551.

7 When you have selected the performance counters for the Siebel measurements you are monitoring, click **OK** to close the Siebel Server Monitor Configuration dialog box. The **Select Measurements to Monitor**

dialog box appears with the selected Siebel measurements in the **Selected Measurements** pane.

8 Click **OK** in the **Select Measurements to Monitor** dialog box, and click **OK** in the Monitors Configuration dialog box, to activate the Siebel Server Manager monitor.

Note: For troubleshooting tips and limitations, see "Troubleshooting Server Resource Monitors," on page 645.

Siebel Server Manager Performance Counters

The following table shows the default counters that can be measured:

Measurement	Description
Average Connect Time	The average connection time.
Average Reply Size	The average size of a user reply.
Average Request Size	The average size of a user request.
Average Requests Per Session	The average number of user requests per session.
Average Response Time	The average amount of time that it takes the server to respond to a request.
Average Think Time	The average amount of think time taken to respond to a request.
Avg SQL Execute Time	The average SQL execute time.
Avg SQL Fetch Time	The average SQL fetch time.
Avg SQL Parse Time	The average SQL parse time.
CPU Time	The CPU time used in the work process.
Elapsed Time	The total amount of elapsed time.

Measurement	Description
Num of DBConn Retries	The number of database connection retries.
Num of DLRbk Retries	The number of DLRbk retries.
Num of Exhausted Retries	The total number of retries that expired.
Number of SQL Executes	The total number of SQL executes.
Number of SQL Fetches	The total number of SQL fetches.
Number of SQL Parses	The total number of SQL parses.
Number of Sleeps	The number of sleeps.
Object Manager Errors	The total number of object manager errors.
Reply Messages	The total number of reply messages.
Request Messages	The total number of request messages.
SQL Execute Time	The total SQL execute time.
SQL Fetch Time	The total SQL fetch time.
SQL Parse Time	The total SQL parse time.
Sleep Time	The total sleep time.
Tests Attempted	The number of tests attempted.
Tests Failed	The number of tests that failed.
Tests Successful	The number of tests that were successful.
Total Reply Size	The total reply size, measured in bytes.
Total Request Size	The total request size, measured in bytes.
Total Response Time	The total response time.

Measurement	Description
Total Tasks	The total number of tasks.
Total Think Time	The total think time.

Configuring the PeopleSoft (Tuxedo) Monitor

The PeopleSoft (Tuxedo) monitor allows you to measure and view the performance of your PeopleSoft (Tuxedo) application server.

If Tuxedo 7.1 or later is installed, you can monitor more than one PeopleSoft (Tuxedo) application server at a time. However, if Tuxedo 6.5 or earlier is installed on the machine, you can monitor only one PeopleSoft (Tuxedo) application server at a time.

Before Configuring the Monitor

Before you set up the monitor, perform the following:

➤ Ensure that a Tuxedo workstation client (not a native client) is installed on the LoadRunner Tuning Module Console machine. Use a Tuxedo 6.x client if a Tuxedo 6.x server is used, and Tuxedo 7.1 or later client if a Tuxedo 7.1 or later server is used. If you use a Tuxedo 6.5 or earlier server, you can still use a Tuxedo 7.1 or later client in order to monitor it, provided that you set the WSINTOPPRE71 environment variable to “yes”.

Note: A Tuxedo workstation client communicates with the application server over the network, and is not required to run the Tuxedo application server on the same machine. A native client can only communicate with the Tuxedo application server if it is part of the relevant Tuxedo domain.

➤ Define the Tuxedo environment variables on the LoadRunner Tuning Module Console machine—set the TUXDIR variable to the Tuxedo installation directory, (for example, V:\environ\32\Tuxedo8.0), and add the Tuxedo bin directory to the PATH variable.

➤ Ensure that the workstation listener (WSL) process is running. This enables the application server to accept requests from workstation clients. Note that the address and port number used to connect to the application server must match those dedicated to the WSL process.

Note: For information on configuring the WSL, refer to the BEA Tuxedo Web site (http://edocs.beasys.com/tuxedo/tux81/rf5/rf5101.htm#1534543).

To configure the PeopleSoft (Tuxedo) monitor:

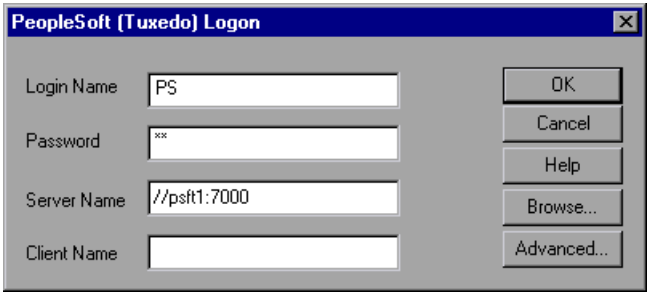
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running PeopleSoft.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **PeopleSoft (Tuxedo)**, and click **Add**.

The PeopleSoft (Tuxedo) Logon dialog box opens.



5 Enter the Tuxedo server **Login Name**, **Password**, **Server Name**, and **Client Name**. If a Tuxedo server was previously monitored, its name is displayed in the Server Name box.

The format of the server name is //<machine name>:<port number>. Alternatively, you can specify the IP address or the hexadecimal format used by old versions of Tuxedo. Note that you cannot use quotation marks.

The default logon values are as follows:

- **Login Name**: PS
- **Password**: PS
- **Port**: 7000

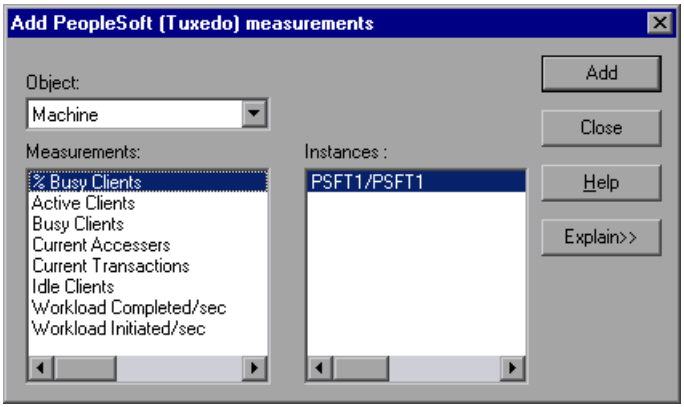
Note: If you are using PeopleSoft 7.x, you can determine the logon information from the *Logon* section of the **tpinit.ini** file in the recorded script's directory.

To authenticate the PeopleSoft (Tuxedo) monitor, click **Advanced**, and enter the authentication data as a hexadecimal string (beginning with "0x") in the data box. The authentication data value can be obtained from the **tpinit.ini** file of an existing Tuxedo script.

Note: If you are using Tuxedo 6.5 or below, the monitor can only connect to one application server during a LoadRunner Tuning Module session step. Once it connects to an application server, that server is the only one used by the monitor until the LoadRunner Tuning Module Console is closed. This applies even when all of the counters are deleted from the monitor.

---

6 Click **OK**. The Add PeopleSoft (Tuxedo) Measurements dialog box opens.



7 Select a PeopleSoft (Tuxedo) object from the **Object** list. Select the measurements and instances you want to monitor. For a list of the available performance counters, see page 557.

8 Click **Add** to place the selected object on the resource list. Add all the desired objects to the list, and click **Close**.

9 Click **OK** in the PeopleSoft (Tuxedo) dialog box.

The Select Measurements to Monitor dialog box appears with the selected PeopleSoft (Tuxedo) measurements in the Selected Measurements pane.

10 Click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the PeopleSoft (Tuxedo) monitor.

PeopleSoft (Tuxedo) Monitor Measurements

The following table describes the default counters that can be measured. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for the APPQ/PSAPPSRV queue.

Monitor	Measurements
Machine	**% Busy Clients** - The per cent of active clients currently logged in to the Tuxedo application server which are waiting for a response from the application server.
	Active Clients - The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients - The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers - Number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions - Number of in use transaction table entries on this machine.
	Idle Clients - The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second - The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second - The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers** - The per cent of active servers currently handling Tuxedo requests.
	Active Servers - The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers - The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers - The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued - The total number of messages which have been placed on the queue.
Server	**Requests/second** - How many server requests were handled per second
	Workload/second -The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec** - The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec - The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec - The number of messages received by the workstation handler, per second.
	Messages Sent/sec - The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec - The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

PeopleSoft Ping Monitoring

The PeopleSoft Ping script monitor displays the data that a user sees on the PeopleSoft Ping page in the *PeopleTools\utilities* folder of the PeopleSoft 8.42 installation. This data is sampled by using a canned lightweight monitoring script, and displayed by the LoadRunner Tuning Module in the PeopleSoft Ping graph.

Note: The PeopleSoft Ping default sampling rate is every five seconds. To change the default sampling rate, select **After the previous iteration ends** in the **General: Pacing** section of the run-time settings, and enter a sampling rate.

To obtain data on the PeopleSoft server:

1. Define the PeopleSoft 8.42 server in the System Topology window.

   For more information on defining system topology, see Chapter 5, "Creating a Topology."

2. Add a step to your session.

   In the **Select Scripts from Template** section, choose **PeopleSoft Ping** from the **PeopleSoft** canned scripts.

   In the **Execute script against the following servers** section, select the PeopleSoft servers against which you want the script to run.

   In the **Script Parameters** section, specify the server, username, and password where PeopleSoft is running.

3. Add the scripts (that you want to run) to the step that already contains the PeopleSoft Ping script.

   In the **Add Step** dialog box, click the **Add...** button to the right of the Custom Scripts pane, add the PeopleSoft scripts that you want to use, and click **OK**. The script names are added to the Custom Scripts pane.

4. Assign one Vuser to run in this script. If you set more than one Vuser, you create unnecessary load on the server.

In the **Add Step** dialog box, click the **Add and Configure** button. The new step is displayed in the session tree using the **ps84_ping_wrapper** script, and the Step Configuration dialog box opens. Enter "1" in the Number of Vusers box, and click **Close**.

5 Execute the session step against a PeopleSoft 8.4 server.

The x-axis represents the time that has elapsed since the start of the session step run. The y-axis represents the resource usage.

37

Java Performance Monitoring

During a session step run, you can monitor the resource usage of Java 2 Platform, Enterprise Edition (J2EE) objects, Enterprise Java Bean (EJB) objects, and Java-based applications using the Java performance monitors.

This chapter describes:

- EJB Performance Monitoring

Note: The J2EE performance monitor is described separately in Chapter 38, "J2EE Performance Monitoring."

About Java Performance Monitoring

The Java performance monitors provide you with performance information for Java 2 Platform, Enterprise Edition (J2EE) objects, Enterprise Java Bean (EJB) objects, and Java-based applications during session step execution. To obtain this data, you need to activate the Java performance monitors before executing the session step, and indicate which statistics and measurements you want to monitor.

EJB Performance Monitoring

You can monitor Enterprise Java Bean (EJB) objects on a WebLogic, WebSphere, or Oracle 9iAS application server during a session step run using the EJB performance monitor.

Support Matrix:

Application Server	Version	Platform
WebLogic	4.x; 5.1; 6.0; 6.1; 7.0	Windows; Solaris; AIX
WebSphere	3.x; 4.x	Windows; Solaris; AIX
Oracle 9i	1.0.2.2	Windows; Solaris; AIX

To monitor EJB objects, you must first install the EJB monitor, run the monitor detector, and activate the EJB monitor on the application server machine. You then configure the EJB monitor on the client machine by selecting the counters you want the monitor to measure.

Note: The server side installation contains new EJBDetector support files for generating EJB Vuser scripts. For more information on the EJBDetector, refer to the *Creating Vuser Scripts* guide.

Installing the EJB Monitor and Running the Monitor Detector

Before EJB objects can be monitored, you must install the EJB monitor support files, and verify that you have a valid JDK environment on the application server machine. You then prepare the EJB monitor for monitoring by running the monitor detector from the batch file, or from the command line.

To install the EJB monitor support files:

Create a home directory for the Mercury Interactive EJB support files—for example, MERC_MONITOR_HOME—and unzip the *<LoadRunner Tuning Module CD>add-ins\Monitors\J2EE\Windows\jmonitor_<platform>.jar* file into that directory.

On UNIX platforms, use the jar utility to extract the installation jar:

Change to the MERC_MONITOR_HOME directory and type the following command:

```
jar -xvf <path to your jmonitor_<platform>.jar>
```

To run the monitor detector from the batch file:

1 Open the *env.cmd* (NT) or *env.sh* (UNIX) file and set the following variables:

JAVA_HOME	Specify the root directory of the JDK installation.
APP_SERVER_DRIVE	Specify the drive on which the application server is installed (for NT only).
DETECTOR_INS_DIR	Specify the root directory of the Detector installation.
APP_SERVER_ROOT	Follow these guidelines: **BEA WebLogic Servers 4.x and 5.x:** Specify the application server root directory. **BEA WebLogic Servers 6.x and 7.x**: Specify the full path of the domain folder. **WebSphere Servers 3.x and 4.0:** Specify the application server root directory. **Oracle OC4J:** Specify the application server root directory. **Sun J2EE Server:** Specify the full path to the deployable *.ear* file or directory containing a number of *.ear* files.
EJB_DIR_LIST (optional)	Specify a list of directories/files, separated by ';' and containing deployable *.ear/.jar* files, and any additional classes directory or *.jar* files or used by your EJBs under test.

2 Run the *Mon_Detector.cmd* (NT) or *Mon_Detector.sh* (UNIX) batch file to collect information about the EJBs deployed. Running the monitor detector generates the following three files in the <*MERC_MONITOR_HOME*>\dat directory: *ejb_monitor.hooks*; *cjhook.ini*; and *regmon.properties*. These files contain information about the EJBs detected on the application server.

Note: You must run the monitor detector each time you add, change, or delete EJBs on the application server.

To run the monitor detector from a command line:

1 Add <*MERC_MONITOR_HOME*>*classes*, <*MERC_MONITOR_HOME*>*dat*, and the <*MERC_MONITOR_HOME*>*classes**xerces.jar* file to the CLASSPATH environment variable.

2 Use the java MonDetect <*search root dir*> command line to collect information about the EJBs deployed.

<*search root dir*>	Specify one or more directories or files in which to search for EJBs (separated by semicolons). Follow these guidelines: **BEA WebLogic Servers 4.x and 5.x:** Specify the application server root directory. **BEA WebLogic Servers 6.x and 7.x**: Specify the full path of the domain folder followed by the root directory. **WebSphere Servers 3.x and 4.0:** Specify the application server root directory. **Oracle OC4J:** Specify the application server root directory. **Sun J2EE Server:** Specify the full path to the deployable *.ear* file or directory containing a number of *.ear* files.

Note that you can also specify a search list of directories and/or files to search. If unspecified, the CLASSPATH will be searched.

Running the monitor detector generates the following three files in the <*MERC_MONITOR_HOME*>\dat directory: *ejb_monitor.hooks*; *cjhook.ini*; and *regmon.properties*. These files contain information about the EJBs detected on the application server.

Note: You must run the monitor detector each time you add, change, or delete EJBs on the application server.

Configuring the EJB Monitor on the Application Server

After you have installed Mercury Interactive's EJB monitor support files on your WebLogic, WebSphere, or Oracle 9iAS machine, you must configure the application server to run with EJB monitor support.

Note: It is important to set the environment variables in the order in which they appear below.

WebLogic Server

The WebLogic 4.x-5.x server, WebLogic 6.x server, and WebLogic 7.x server are configured differently.

To configure the WebLogic 4.x-5.x server:

1. Copy the <*WebLogic Home*>*startWeblogic.cmd* file into <*WebLogic Home*>*startWeblogicMercury.cmd* so that the file is backed up.
2. Open the <*WebLogic Home*>*startWeblogicMercury.cmd* file.
3. In the 'runWebLogicJava' section of the file, after the WEBLOGIC_CLASSPATH environment settings, set the following environment variables:

   For Windows platforms:

   ```
   set MERC_MONITOR_HOME=<EJB Monitor Home Directory>

   set CLASSPATH=%MERC_MONITOR_HOME%\dat

   set JAVA_CLASSPATH=%MERC_MONITOR_HOME%\dat;%MERC_
   MONITOR_HOME%\classes;%MERC_MONITOR_HOME%\classes\
   xerces.jar;%JAVA_CLASSPATH%
   ```

```
set PATH=%PATH%;%MERC_MONITOR_HOME%\bin
```

For UNIX platforms:

```
MERC_MONITOR_HOME <EJB Monitor Home Directory>

CLASSPATH ${MERC_MONITOR_HOME}/dat

JAVA_CLASSPATH ${MERC_MONITOR_HOME}/dat:${MERC_MONITOR_
HOME}/classes:${MERC_MONITOR_HOME}/classes/xerces.jar:${JAVA_
CLASSPATH}

LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${MERC_MONITOR_HOME}/bin

export CLASSPATH

export LD_LIBRARY_PATH

export JAVA_CLASSPATH
```

Note: For IBM AIX platform replace LD_LIBRARY_PATH with LIBPATH. Replace <*EJB Monitor Home Directory*> with the EJB monitor installation root directory. Note that on UNIX platforms you may have to export the library path variables.

---

4 In the same section of the file, add a parameter to the command line:

```
-Xrunjdkhook.
```

For example on Windows platforms:

```
%JAVA_HOME%\bin\java -ms64m -mx64m -Xrunjdkhook -classpath
%JAVA_CLASSPATH%  -Dweblogic.class.path=%WEBLOGIC_CLASSPATH%
-Dweblogic.home=. -Djava.security.manager
-Djava.security.policy==.\weblogic.policy weblogic.Server
```

Note: For Solaris installation only.
If you are using JDK 1.2.x add a parameter to the command line:
-Dweblogic.classloader.preprocessor=com.mercuryinteractive.aim.
MercuryWL5Preprocessor

for example, on Windows platforms:
%JAVA_HOME%\bin\java -ms64m -mx64m -classpath %JAVA_CLASSPATH%
-Dweblogic.classloader.preprocessor=com.mercuryinteractive.aim.
MercuryWL5Preprocessor
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH%
-Dweblogic.home=. -Djava.security.manager
-Djava.security.policy==.\weblogic.policy weblogic.Server

---

5 Run the <*WebLogic Home*>*startWeblogicMercury.cmd* file.

To configure the WebLogic 6.x server:

1 Copy the <*WebLogic Home*>*config*\<*domain name*>*startWeblogic.cmd* file into <*WebLogic Home*>*config*\<*domain name*>*startWeblogicMercury.cmd* so that the file is backed up.

2 Open the <*WebLogic Home*>*config*\<*domain name*>*startWeblogicMercury.cmd* file.

3 In the 'runWebLogic' section of the file, set the following environment variables:

For Windows platforms:

set MERC_MONITOR_HOME=<your MERC_MONITOR_HOME directory>

set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;%
MERC_MONITOR_HOME%\classes;%MERC_MONITOR_HOME%\classes\
xerces.jar

set PATH=%PATH%;%MERC_MONITOR_HOME%\bin

For UNIX platforms:

```
MERC_MONITOR_HOME <EJB Monitor Home Directory>

CLASSPATH ${JAVA_CLASSPATH}:${MERC_MONITOR_HOME}/dat:$
{MERC_MONITOR_HOME}/classes:${MERC_MONITOR_HOME}/classes/
xerces.jar

LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${MERC_MONITOR_HOME}/
bin

export CLASSPATH

export LD_LIBRARY_PATH
```

Note: For IBM AIX platform replace LD_LIBRARY_PATH with LIBPATH. Replace <*EJB Monitor Home Directory*> with the EJB monitor installation root directory. Note that on UNIX platforms you may have to export the library path variables.

---

4 In the same section of the file add a parameter to the command line:

```
-Xrunjdkhook.
```

for example, on Windows platforms:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m -Xrunjdkhook -classpath
%CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea"
“-Djava.security.policy==f:\bea\wlserver6.0/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

5 Run the <*WebLogic Home*>*config*\<*domain name*>\ *startWeblogicMercury.cmd* file.

To configure the WebLogic 7.x server:

1. Copy the <*WebLogic Home*>*server**bin**startwls.cmd* file into <*WebLogic Home*>*server**bin**startwlsMercury.cmd* so that the file is backed up.
2. Open the <*WebLogic Home*>*server**bin**startwlsMercury.cmd* file.
3. In the 'runWebLogic' section of the file, set the following environment variables:

   For Windows platforms:

   ```
   set MERC_MONITOR_HOME=<your MERC_MONITOR_HOME directory>

   set
   CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;%MERC_M
   ONITOR_HOME%\classes;%MERC_MONITOR_HOME%\classes\xerces.jar

   set PATH=%PATH%;%MERC_MONITOR_HOME%\bin
   ```

   For UNIX platforms:

   ```
   MERC_MONITOR_HOME <EJB Monitor Home Directory>

   CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:$MERC_
   MONITOR_HOME/classes:$MERC_MONITOR_HOME/classes/xerces.jar

   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$MERC_MONITOR_HOME/bin

   export CLASSPATH

   export LD_LIBRARY_PATH
   ```

Note: For IBM AIX platform replace LD_LIBRARY_PATH with LIBPATH. Replace <*EJB Monitor Home Directory*> with the EJB monitor installation root directory. Note that on UNIX platforms you may have to export the library path variables.

---

4 In the same section of the file add a parameter to the command line:

-Xrunjdkhook.

for example, on Windows platforms:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m -Xrunjdkhook -classpath
%CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea"
“-Djava.security.policy==f:\bea\wlserver6.0/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

5 Copy the *<domain name>\startWeblogic.cmd* file into *<domain name>\startWeblogicMercury.cmd* so that the file is backed up.

6 Open the *<domain name>\startWeblogicMercury.cmd* file.

7 Find the call to the weblogic server. For example, call:
D:\bea\weblogic700\server\bin\startWLS.cmd

8 Change the call from *startWLS.cmd* to *startWLSMercury.cmd*, and save the file.

9 Run the *<domain name>\startWeblogicMercury.cmd* file.

WebSphere Server - Versions 3.0 and 3.5

By default, the WebSphere 3.x application server runs as an automatic service, upon machine startup. Since Mercury Interactive does not currently support LoadRunner Tuning Module EJB monitoring on a WebSphere server run as an automatic service, you must change the default WebSphere server startup to *manual*.

To change the default WebSphere 3.x server startup:

1 Select **Start** > **Settings** > **Control Panel**.

2 Double-click **Services**.

3 Select **IBM WS AdminServer**, and click the **Stop** button.

4 Double click **IBM WS AdminServer**, and select the **Manual** Startup Type.

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

You can now start the WebSphere Server from <*WebSphere Home*>\AppServer\bin\debug\adminserver.bat, instead of using the automatic service.

To add LoadRunner Tuning Module EJB monitor support to the WebSphere 3.x server:

1. Make a backup copy of the <*WebSphere Home*>*AppServer**bin**debug**adminserver.bat* file.
2. Open the <*WebSphere Home*>*AppServer**bin**debug**adminserver.bat* file.
3. Add the following environment variables at the end of the 'SET_CP' section:

   For Windows platforms:

   ```
   set CLASSPATH=<MERC_MONITOR_HOME>\dat;<MERC_MONITOR_
   HOME>\classes;<MERC_MONITOR_HOME>\classes\xerces.jar;
   %CLASSPATH%

   set PATH=%PATH%;<MERC_MONITOR_HOME>\bin
   ```

   For UNIX platforms:

   ```
   CLASSPATH ${MERC_MONITOR_HOME}/dat:${MERC_MONITOR_HOME}/
   classes:${MERC_MONITOR_HOME}/classes/xerces.jar:${CLASSPATH}

   LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${MERC_MONITOR_HOME}/bin

   export CLASSPATH

   export LD_LIBRARY_PATH
   ```

Note: For IBM AIX platform replace LD_LIBRARY_PATH with LIBPATH. Replace <*EJB Monitor Home Directory*> with the EJB monitor installation root directory. Note that on UNIX platforms you may have to export the library path variables.

Note: For Solaris installation only.
If you are working with JRE1.2.x, you must download the patch file, PQ46831.jar, from IBM's Web site or FTP site:
http://www-3.ibm.com/software/webservers/appserv/efix-archive.html
ftp://ftp.software.ibm.com/software/websphere/appserv/support/fixes/pq46831/
Make sure to download the version that corresponds to your server version.
Add the patch file to the classpath:
setenv CLASSPATH PQ46831.jar:${CLASSPATH}

---

4 Run the *adminserver.bat* file.

5 Open the WebSphere Advanced Administrative Console, and select **View** > **Topology**.

6 Expand the WebSphere Administrative Domain tree by selecting <**server machine name**> > **Default Server**.

7 Select the **General** tab in the Application Server:Default Server window.

8 Type -Xrunjdkhook in the command line Arguments box, and click **Apply**.

If you are working with a WebSphere 3.0 Server with JDK1.1.7 IBM, double-click on **Environment**. Type _CLASSLOAD_HOOK in the Variable Name box, and jdkhook in the Value box. Click the **Add**, **OK**, and **Apply** buttons.

Note: For Solaris installation only.
If you are working with a WebSphere 3.5 Server with J2RE1.2.x, in the Command Line Arguments box, type the following and click **Apply**:
-Dcom.ibm.ejs.sm.server.ServiceInitializer=com.ibm.ejs.sm.server.WilyInitializer
-Dcom.ibm.websphere.introscope.implClass=com.mercuryinteractive.aim.MercuryWASPreprocessor

---

9 Close the WebSphere Advanced Administrative Console.

10 Close and restart the *adminserver.bat* file.

WebSphere Server - Version 4.0

You can start the WebSphere 4.0 server using the startServerBasic.bat file or the startServer.bat file.

To configure the WebSphere 4.0 server:

1. Ensure that the WebSphere Administrative Server is running, and start the Administrator Console.
2. In the WebSphere Administrative Domain tree, expand the Nodes, hostname, and Application Servers subtrees, and select the Default Server (or the Application Server you wish to use with JMonitor).
3. For Windows 2000/NT or Solaris, click the **General** tab, and add the following variables to the **Environment** box:

Note: Replace *<EJB Monitor Home Directory>* with the EJB monitor installation root directory.

For Windows 2000/NT:

name=PATH

value=<*EJB Monitor Home Directory*>\bin

For Solaris:

name=LD_LIBRARY_PATH

value=<*EJB Monitor Home Directory*>/bin

Click **OK** to close the **Environment Editor** dialog box.

For AIX:

If the LIBPATH environment variable has been changed, you need to link the EJB monitor libraries to the /usr/lib directory.

Add the following command:

#ln -s <*EJB Monitor Home Directory*>/bin/libcjhook_mon.so /usr/lib/libcjhook_mon.so

#ln -s <*EJB Monitor Home Directory*>/bin/libconfig.so /usr/lib/libconfig.so

#ln -s <*EJB Monitor Home Directory*>/bin/libjdkhook.so /usr/lib/libjdkhook.so

#ln -s <*EJB Monitor Home Directory*>/bin/libmlib_ds.so /usr/lib/libcjhook_mon.so

#ln -s <*EJB Monitor Home Directory*>/bin/libmosifs.so /usr/lib/libmosifs.so

#ln -s <*EJB Monitor Home Directory*>/bin/libthrdutil.so /usr/lib/libthrdutil.so

---

Note: You will likely require root permissions in order to create the link. Alternatively, you can place the link in WebSphere's /bin directory (usually /usr/WebSphere/AppServer/bin).

---

4 Click the **JVM Settings** tab in the WebSphere Administrative Console, and add the following values to the classpath:

Note: Replace <*EJB Monitor Home Directory*> with the EJB monitor installation root directory.

For Windows 2000/NT:

<*EJB Monitor Home Directory*>\dat

<*EJB Monitor Home Directory*>\classes

<*EJB Monitor Home Directory*>\classes\xerces.jar

For Solaris or AIX:

<*EJB Monitor Home Directory*>/dat

<*EJB Monitor Home Directory*>/classes

<*EJB Monitor Home Directory*>/classes/xerces.jar

Note: For Solaris installation only.
If you are working with JRE1.2.x, you must download the patch file, PQ46831.jar, from IBM's Web site or FTP site:
http://www-3.ibm.com/software/webservers/appserv/efix-archive.html
ftp://ftp.software.ibm.com/software/websphere/appserv/support/fixes/pq46831/
Make sure to download the version that corresponds to your server version.
Add the following value to the classpath:
<EJB Monitor Home Directory>/classes/PQ46831.jar

---

5 Click the **Advanced JVM Settings** button. In the Command line arguments field, add the following value for Windows 2000/NT, Solaris, and AIX:

-Xrunjdkhook

Note: For Solaris installation only.
If you are working with JRE1.2.x, instead of -Xrunjdkhook
add the following value:
-Dcom.ibm.ejs.sm.server.ServiceInitializer=com.ibm.ejs.sm.server.WilyInitializer
-Dcom.ibm.websphere.introscope.implClass=com.mercuryinteractive.aim.MercuryWASPreprocessor

---

6 Click the **OK** and **Apply** buttons to save the changes for the Application server. You can now start and stop your WebSphere server using the LoadRunner Tuning Module EJB Monitor.

Oracle 9iAS Server

Once you have configured the support files and set up the JDK environment on the Oracle 9iAS application server, run the *oc4jMonitor.cmd* file on an NT machine, or the *oc4jMonitor.sh* file on a UNIX machine. The application server starts running with EJB monitor support.

Configuring the EJB Monitor on the Client Machine

To monitor EJB performance, you must select the counters you want the EJB monitor to measure. You select these counters using the Console's EJB Monitor Configuration dialog box.

To configure the EJB monitor:

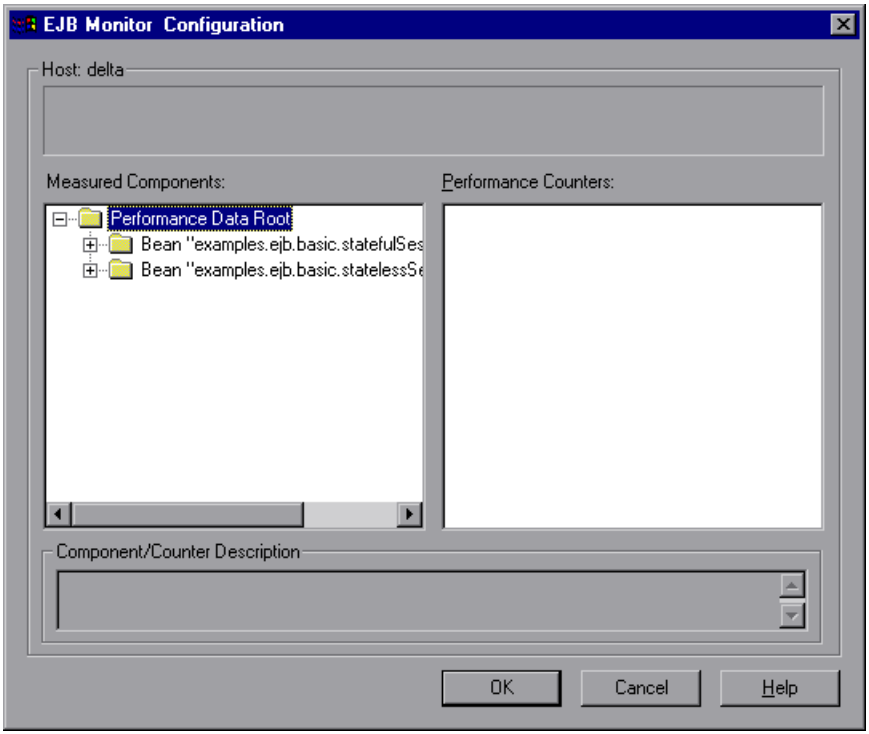
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Java Technology category, select **EJB**, and then click **Add**.

The EJB Monitor Configuration dialog box opens, displaying the available EJBs.



5 Expand the Measured Components tree and select the methods and counters you want to monitor.

For a list of the available performance counters, see page 578.

6 Click **OK** in the EJB Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the EJB monitor.

EJB Performance Counters

The following counters can be monitored for each method:

Measurement	Description
Average Response Time	The average response time, in milliseconds, of the EJB object being monitored.
Method Calls per Second	The number of EJB object method calls per second.

38

J2EE Performance Monitoring

The J2EE performance monitor provides complete insight into the J2EE components on the application server (Servlets, JSP's, EJB's, JNDI, JDBC, and DB SQL calls).

This chapter describes:

- Installing the J2EE Monitor on the Application Server
- Initial J2EE Monitor Configuration Settings
- Activating the J2EE Monitor on the Client Machine
- Examples of Modifying Application Server Configurations
- Troubleshooting the J2EE Monitor

About J2EE Performance Monitoring

The J2EE monitor provides the following information for each J2EE component:

- Average response time per method/query
- Number of method calls per second

With such coverage of the J2EE architecture, users can get an overview of the entire activity within the system. They can very easily correlate the end user response time with the Web server activity (Servlets and JSPs data), application server activity (JNDI and EJB's), and back-end activity of database requests (JDBC methods and SQL queries).

The J2EE Monitor allows LoadRunner Tuning Module users to analyze J2EE component metrics during a session step run by using an agent which is installed on the application server to collect information on the J2EE components. These measurements are sent from the application server back to the LoadRunner Tuning Module Console through a Web server contained in the J2EE monitor. The J2EE Monitor supports the leading applications servers, such as: IBM WebSphere, BEA WebLogic, Oracle 9iAS and JBoss. For information about the supported application servers, refer to the "Support Matrix" on page 581.

Note: The J2EE Monitor requires MSXML 3.0 and later (this is included in Internet Explorer 6.0). You can install MSXML 3.0 from the Microsoft MSDN Web site (http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/msdn-files/027/001/772/msdncompositedoc.xml).

Installing the J2EE Monitor on the Application Server

In order to monitor J2EE objects, you must first install and activate the J2EE monitor on the application server machine. You then configure the J2EE monitor on the client machine by selecting the counters you want the monitor to measure.

You can monitor Java 2 Platform, Enterprise Edition (J2EE) objects on a WebLogic, WebSphere, Oracle 9iAS, or JBoss application server during a session step run using the J2EE performance monitor.

Support Matrix

Application Server	Version	Platform
WebLogic	4.x; 5.x; 6.x; 7.0; 8.1	Windows; Solaris; AIX
WebSphere	3.x; 4.x	Windows; Solaris; AIX
Oracle 9iAS	1.0.2.2	Windows; Solaris; AIX
JBoss	2.4.x; 3.04	Windows; Solaris; AIX

To install the J2EE monitor on the application server:

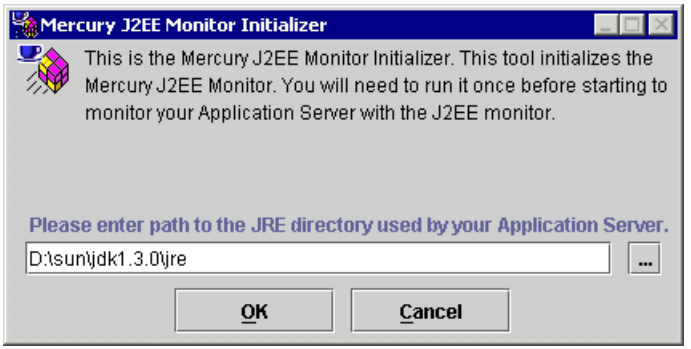
1 Create a home directory on the application server machine—for example, *J2EEMonitor*, and unzip the installation file *<LoadRunner Tuning Module CD>\Add-ins\J2EE\jmonitor_<platform>.jar* file into that directory.

If you do not have WinZip to unzip the installation file, use the following command line to extract the installation file:

```
<JDK>\bin\jar.exe -xf <installation file>
```

UNIX scripts extracted from the jar file may lose their execute permissions. To fix this, open the J2EEMonitor Home Directory, and change the permissions using the command line: `chmod +x *.sh`.

2 Open the Mercury J2EE Monitor Initializer by double-clicking *<J2EEMonitor Home Directory>\classes\sipatool.jar*.



On a UNIX platform, or if the *.jar* extension in your system is not associated with the Java runtime environment, run *sipatool.jar* as follows:

From the *<J2EEMonitor Home Directory>\classes* directory, type java -jar sipatool.jar.

Note: If the Mercury J2EE Monitor Initializer is not displayed, run *sipatool.jar* as follows:

From the *<J2EEMonitor Home Directory>\classes* directory, type <Java path>\ -jar sipatool.jar -nogui. Use the full path of the Java process of the application server JDK.

---

3 In the Mercury J2EE Monitor Initializer, enter the path to the application server Java home directory, and click **OK** to run the tool.

4 Add -Xbootclasspath/p:<J2EEMonitor Home Directory>\classes\boot to the application server command line arguments.

Refer to "Examples of Modifying Application Server Configurations", on page 588 to see syntax for WebLogic, WebSphere, Oracle 9iAS, or JBoss application servers.

Initial J2EE Monitor Configuration Settings

The J2EE monitor application server installation configured the hooking mechanism, operation mode, JDBC, and EJB information retrieval.

Hooking mechanism: The J2EE monitor uses the Mercury J2EE Monitor Initializer and Java hooking library.

Operation mode: The J2EE monitor uses the Auto Discovery operating mode. In this mode, the system automatically discovers the J2EE components (Servlet, JSP, JNDI, EJB and JDBC) that actually participate in the business process.

JDBC information retrieval: The JDBC information retrieval setting determines which data to return from the JDBC call. By default, the J2EE monitor aggregates the measured data according to the JDBC operation, for example: SELECT,UPDATE,CREATE. To modify this configuration, refer to "Configuring JDBC Information Retrieval" on page 584.

EJB information retrieval: The EJB information retrieval setting determines which data to return from the EJB call. By default, the J2EE monitor is not configured to measure container methods, (e.g., ejbPassivate(), ejbCreate()). To modify this configuration, refer to "Configuring the EJB Information Retrieval" on page 584.

Note: For information about alternative configuration settings, please contact Mercury Interactive Customer Support.

Configuring JDBC Information Retrieval

1 Open *<J2EEMonitor Home Directory>\dat\monitor.properties*.

2 In the property monitor.jdbc.mode, enter one of the following:

- "1" to measure the JDBC the method calls, like any other (non-JDBC) measured method calls.
- "2" to aggregate the measured data according to the JDBC operation, for example: SELECT,UPDATE,CREATE.
- "3" to aggregate the measured data according to specific SQL statement (including the operation, the table(s) it acted on, and other parameters of this statement).

Note: SQL Statements that exceed 3000 characters in length are not supported.

Configuring the EJB Information Retrieval

1 Open <J2EEMonitor Home Directory>\dat\java_monitor.ini.

2 In the "EJB_CONFIG" section of the file, set the following property:

hook_files to hook_files=auto_detect_container.

Activating the J2EE Monitor on the Client Machine

To monitor J2EE performance, you must select the counters you want the J2EE monitor to measure. You select these counters using the Console's J2EE Monitor Configuration dialog box.

Before configuring the J2EE monitor:

In Auto Discovery mode (the J2EE monitor's default operating mode), the system discovers which methods of the components (Servlet, JSP, JNDI, EJB and JDBC) are participating in your business process and measures those objects only.

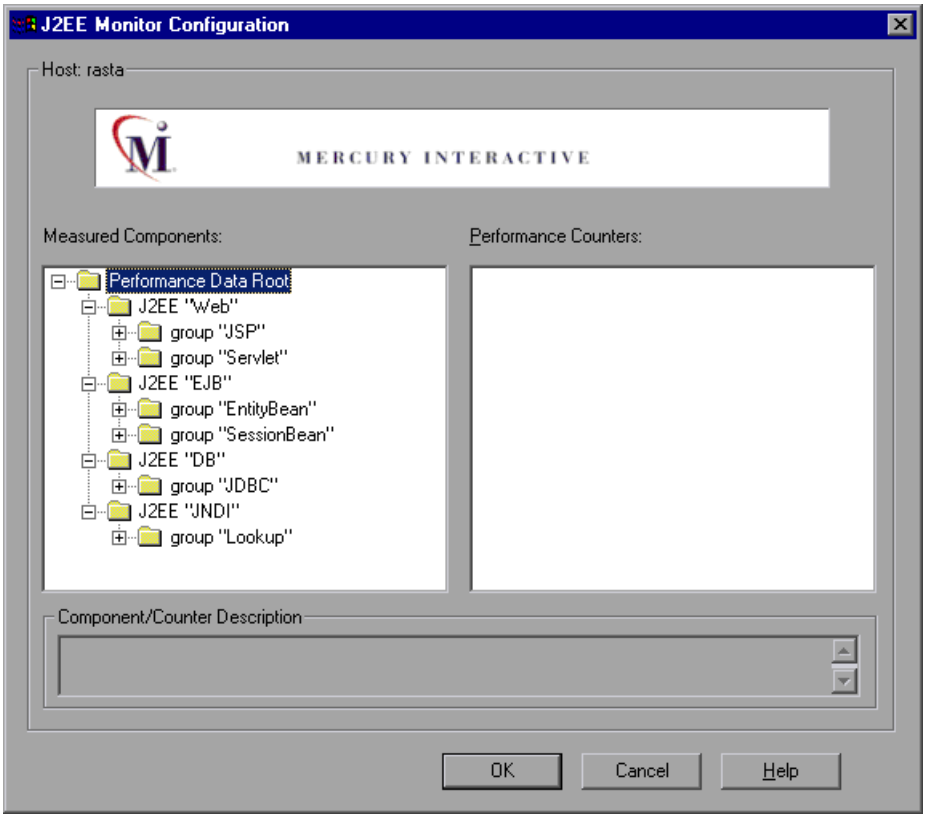
To start the Auto Discovery process, start the application server, and run the Vuser script that you intend to use in your load test against the application server. This provides the Console with a list of measurements that will be available for monitoring.

Note: The next time you run the same script, you don't need to run a Vuser before selecting the methods and counters you want to monitor.

To configure the J2EE monitor:

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Java Technology category, select **J2EE**, and then click **Add**.

The J2EE Monitor Configuration dialog box opens, displaying the available J2EE counters.



5 Expand the Measured Components tree and select the methods and counters you want to monitor.

For a list of the available performance counters, see page 587.

Note: The size of a measurement name that can be displayed in the Analysis is limited to 255 characters. If a measurement name exceeds this limit, the counter name is truncated, and given a unique ID (UID). If you monitor different events or make cross result graphs on the same counter, the UID will stay the same.

The measurement name is truncated as follows:

standard prefix/**counter truncated name<UID>**/monitored event

For example:

/DB/JDBC/weblogic.jdbc.rmi.SerialPreparedStatement/int executeUpdate()/**INSERT INTO orders (orderid _ userid _ orderdate _ shipaddr1 _ shipaddr2 _ shipcity _ shipstate _ shipzip _ shipcountry _ billaddr1 _ billaddr2 _ b <1>** /Average Response Time

The full measurement name appears in the Measurement Description box.

---

6 Click **OK** in the J2EE Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the J2EE monitor.

J2EE Performance Counters

The following counters can be monitored for each method:

Measurement	Description
Average Response Time	The average response time, in milliseconds, of the J2EE object being monitored.
Method Calls per Second	The number of J2EE object method calls per second.

Examples of Modifying Application Server Configurations

When you installed Mercury Interactive's J2EE monitor files on your application server, you already configured it to run with J2EE monitor support. This section provides examples modifying the configuration of the following application servers:

- WebLogic - Version 4.x-5.x
- WebLogic - Version 6.x
- WebLogic - Version 7.x
- WebLogic - Version 8.1
- WebSphere Server - Version 3.x
- WebSphere Server - Version 4.x
- Oracle 9iAS Server
- JBoss 2.4.x-3.04 Server

Note: It is important to set the environment variables in the order in which they appear below.

WebLogic Server

Each WebLogic server is configured differently.

WebLogic - Version 4.x-5.x

To configure the WebLogic 4.x-5.x server:

1. Copy the <*WebLogic Home*>*startWeblogic.cmd* file into <*WebLogic Home*>*startWeblogicMercury.cmd* so that the file is backed up.
2. Open the <*WebLogic Home*>*startWeblogicMercury.cmd* file.
3. Just before the Java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set JAVA_CLASSPATH=%JAVA_CLASSPATH%;%MERC_MONITOR_HOME
%\dat
```

For UNIX platforms (csh):

```
MERC_MONITOR_HOME <J2EEMonitor Home Directory>
JAVACLASSPATH=$JAVACLASSPATH:$MERC_MONITOR_HOME/dat
```

4 In the same section of the file, add the following parameter to the Java command line: -Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot

Example:

```
%JAVA_HOME%\bin\java -ms64m -mx64m -
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH% -Dweblogic.home=.
-Djava.security.manager
-Djava.security.policy==.\weblogic.policy weblogic.Server
```

5 Run the <*WebLogic Home*>*startWeblogicMercury.cmd* file.

WebLogic - Version 6.x

To configure the WebLogic 6.x server:

1 Copy the <*WebLogic Home*>*config*\<*domain name*>*startWeblogic.cmd* file into <*WebLogic Home*>*config*\<*domain name*>*startWeblogicMercury.cmd* so that the file is backed up.

2 Open the <*WebLogic Home*>*config*\<*domain name*>*startWeblogicMercury.cmd* file.

3 Just before the java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat
```

4 In the same section of the file add a parameter to the command line:

-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot

Example:

5 Run the *<WebLogic Home>\config\<domain name>\startWeblogicMercury.cmd* file.

WebLogic - Version 7.x

To configure the WebLogic 7.x server:

1 Copy the *<WebLogic Home>\server\bin\startwls.cmd* file into *<WebLogic Home>\server\bin\startwlsMercury.cmd* so that the file is backed up.

2 Open the *<WebLogic Home>\server\bin\startwlsMercury.cmd* file.

3 Just before the java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat
```

4 In the same section of the file add a parameter to the command line:

-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot

Example:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-classpath %CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea" "
-Djava.security.policy==f:\bea\wlserver6.0/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

5 Copy the *<domain name>\startWeblogic.cmd* file into *<domain name>\startWeblogicMercury.cmd* so that the file is backed up.

6 Open the *<domain name>\startWeblogicMercury.cmd* file.

7 Find the call to the Weblogic server. For example, call:
D:\bea\weblogic700\server\bin\startWLS.cmd

8 Change the call from *startWLS.cmd* to *startWLSMercury.cmd*, and save the file.

WebLogic - Version 8.1

To configure the WebLogic 8.1 server:

1 On the monitor machine, change the following line in **Monitor Home\dat\modules.properties**:

Change from:
japata.hook.class.name=
com.mercuryinteractive.japata.cjhook_mon.CJHookItJava

To:
japata.hook.class.name=
com.mercuryinteractive.japata.cjhook_mon.CJHookItC

2 Back up the batch file that is used to start the application, located in **<WebLogic Home>\samples\domains\<domain>**. Batch file names begin

with "start" followed by the name of the application (for example, s**tartwlsMercury.cmd** file).

3 In the batch file, just before the java command line used to start the server, add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<Monitor installation>
set CLASSPATH=%CLASSPATH%;
%MERC_MONITOR_HOME%\classes\jdom.jar;
%MERC_MONITOR_HOME%\classes\xalan.jar;
%MERC_MONITOR_HOME%\classes\xerces.jar;
%MERC_MONITOR_HOME%\classes\xml-apis.jar;
%MERC_MONITOR_HOME%\classes;
%MERC_MONITOR_HOME%\classes\sqlkey.jar;
%MERC_MONITOR_HOME%\classes\antlr.jar
set PATH=%PATH%;%MERC_MONITOR_HOME%\bin
```

For UNIX platforms:

```
set MERC_MONITOR_HOME=<Monitor installation>
set CLASSPATH=%CLASSPATH%;
%MERC_MONITOR_HOME%\classes\jdom.jar;
%MERC_MONITOR_HOME%\classes\xalan.jar;
%MERC_MONITOR_HOME%\classes\xerces.jar;
%MERC_MONITOR_HOME%\classes\xml-apis.jar;
%MERC_MONITOR_HOME%\classes;
%MERC_MONITOR_HOME%\classes\sqlkey.jar;
%MERC_MONITOR_HOME%\classes\antlr.jar
set PATH=%PATH%;%MERC_MONITOR_HOME%\bin
```

4 In the same section of the file, add the following parameter to the command line:

```
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot -Xrunjdhook
```

Example:

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS%
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-Xrunjdkhook %JAVA_OPTIONS%
-Dweblogic.Name=%SERVER_NAME%
-Dweblogic.management.username=%WLS_USER%
-Dweblogic.management.password=%WLS_PW%
-Dweblogic.ProductionModeEnabled=%STARTMODE%
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"
weblogic.Server
```

5 Copy <**domain name**>**startWeblogic.cmd** as a new file named <**domain name**>**startWeblogicMercury.cmd**.

6 Open <**domain name**>**startWeblogicMercury.cmd**.

7 Find the call to the Weblogic server. For example:
call D:\bea\weblogic700\server\bin\startWLS.cmd

8 Change the call from **startWLS.cmd** to **startWLSMercury.cmd**, and save the file.

WebSphere Server - Version 3.x

By default, the WebSphere 3.x application server runs on Windows as an automatic service, upon machine startup. Since Mercury Interactive does not currently support LoadRunner Tuning Module J2EE monitoring on a WebSphere server run as an automatic service, you must change the default WebSphere server startup to *manual*.

To change the default WebSphere 3.x server startup:

1 Select **Start** > **Settings** > **Control Panel**.

2 Double-click **Services**.

3 Select **IBM WS AdminServer**, and click the **Stop** button.

4 Double-click **IBM WS AdminServer**, and select the **Manual** Startup Type.

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

You can now start the WebSphere Server from *<WebSphere Home>\AppServer\bin\debug\adminserver.bat*, instead of using the automatic service.

To add LoadRunner Tuning Module J2EE monitor support to the WebSphere 3.x server:

1. Make a backup copy of the *<WebSphere Home>\AppServer\bin\debug\adminserver.bat* file.
2. Open the *<WebSphere Home>\AppServer\bin\debug\adminserver.bat* file.
3. Add the following environment variables at the end of the 'SET_CP' section:

   For Windows platforms:

   ```
   set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
   ```

   For UNIX platforms:

   ```
   MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
   export MERC_MONITOR_HOME
   ```

4. Run the *adminserver.bat* file.
5. Open the WebSphere Advanced Administrative Console, and select **View** > **Topology**.
6. Expand the WebSphere Administrative Domain tree by selecting **<server machine name>** > **Default Server**.
7. Select the **General** tab in the Application Server:Default Server window.
8. Add -Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot to the command line Arguments box, and click **Apply**.

   If you are working with a WebSphere 3.0 Server with JDK1.1.7 IBM, double-click on **Environment**. Type _CLASSLOAD_HOOK in the Variable Name box, and jdkhook in the Value box. Click the **Add**, **OK**, and **Apply** buttons.
9. For Windows 2000/NT or Solaris, open the Environment Editor dialog box from the General tab, and add the following variables to the Environment box:

For Windows 2000/NT:

```
name=CLASSPATH
value=<J2EEMonitor Home Directory>\dat
```

For Solaris:

```
name=CLASSPATH
value=<J2EEMonitor Home Directory>\dat
```

Click **OK** to close the Environment Editor dialog box.

10 Close the WebSphere Advanced Administrative Console.

11 Close and restart the *adminserver.bat* file.

WebSphere Server - Version 4.x

You can start the WebSphere 4.x server using the startServerBasic.bat file or the startServer.bat file.

To configure the WebSphere 4.x server:

1 Ensure that the WebSphere Administrative Server is running, and start the Administrator Console.

2 In the WebSphere Administrative Domain tree, expand the Nodes, Hostname, and Application Servers subtrees, and select the Default Server (or the application server you wish to use with J2EE monitor).

3 Right-click the Default Server, select Properties from the menu, and click the **General** tab.

4 For Windows 2000/NT or Solaris, open the Environment Editor dialog box from the General tab, and add the following variables to the Environment box:

For Windows 2000/NT:

```
name=CLASSPATH
value=<J2EEMonitor Home Directory>\dat
```

For Solaris:

```
name=CLASSPATH
value=<J2EEMonitor Home Directory>\dat
```

Click **OK** to close the Environment Editor dialog box.

5 Click the **Advanced JVM Settings** tab and select **Advanced JVM settings**. In the Command line arguments field, add the following value for Windows 2000/NT, Solaris, and AIX:

```
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
```

6 Click the **OK** and **Apply** buttons to save the changes for the Application server. You can now start and stop your WebSphere server using the LoadRunner Tuning Module J2EE Monitor.

Oracle 9iAS Server

1 Edit the file *env.cmd* (*env.sh* in Unix platforms) as follows:

- the JAVA_HOME environment variable should point to the location of the Java Virtual machine used to run the application server.

- the DETECTOR_INS_DIR environment variable should point to the location of the monitor installation.

- the APP_SERVER_DRIVE environment variable should specify the drive hosting the application server installation (e.g., D:). Do not modify this variable on Unix Platforms.

- the APP_SERVER_ROOT environment variable should specify the application server root directory.

2 Run the *oc4jMonitor.cmd* (*oc4jMonitor.sh* in Unix platforms).

JBoss 2.4.x-3.04 Server

1 Make a backup copy of *<JBoss Home>\run.bat* (run.sh on Unix platforms) file into *<JBoss Home>\runMercury.bat* (*runMercury.sh* for Unix).

2 Open the *<JBoss Home>\runMercury.bat* file (*runMercury.sh* on Unix).

Just before the Java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EE Monitor Home Directory>
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EE Monitor Home Directory>
```

3 In the same section of the file add the following parameter to the command line:

-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot

Example:

```
%JAVA_HOME%\bin\java -ms64m -mx64m  -Xbootclass-
path/p:%MERC_MONITOR_HOME%\classes\boot
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH% -Dweblogic.home=.
-Djava.security.manager
-Djava.security.policy==.\weblogic.policy weblogic.Server
```

4 Run the *<JBoss Home>\runMercury.bat* (*<JBoss Home>\runMercury.sh*) file.

Troubleshooting the J2EE Monitor

Changing the Default Port

The J2EE monitor communicates with LoadRunner Tuning Module, by default, using port 2004. If this port has already been taken, you can select another port as follows:

1. On the application server machine, open *<J2EEMonitor Home Directory>\dat\monitor.properties* and change the port number specified in the property: webserver.monitor.port
2. On the LoadRunner Tuning Module machine, open *<LoadRunner Tuning Module Home Directory>\dat\monitors\xmlmonitorshared.ini* and change the port number specified in section "mon_j2ee" under the key "DefaultPort".

Initialization Errors

If you are getting application server initialization errors such as: "UnsupportedClassVersionError", "NoSuchMethodError" or "NoClassDefFoundError", there might be a conflict between the JDK version specified using the Mercury J2EE Monitor Initializer, and the actual JDK version used in application server launch.

Make sure that you selected the correct JDK that is currently being used by the application server. Note that if you switched the application server to work with a different JDK, you must run the Mercury J2EE Monitor Initializer again.

39

Application Deployment Solution Monitoring

During a session step run, you can monitor the Citrix MetaFrame XP server in order to isolate server performance bottlenecks.

This chapter describes:

➤ Configuring the Citrix MetaFrame Server Monitor

About Application Deployment Solution Monitoring

LoadRunner Tuning Module's Citrix MetaFrame XP monitor provides you with information about the application deployment usage of the Citrix MetaFrame XP server during a session step execution. The Citrix Monitor allows you to monitor the server performance statistics from Citrix Metaframe Servers. You can monitor multiple parameters (counters) with a single monitor instance. This allows you to watch server loading for performance, availability, and capacity planning.

Configuring the Citrix MetaFrame Server Monitor

In order to obtain data for this graph, you need to activate the Citrix MetaFrame server monitor before executing the session step, enable the counters you want to monitor on the Citrix server, and specify which measurements and resources you want the Citrix monitor to measure.

You select measurements to monitor the Citrix MetaFrame server using the Citrix MetaFrame XP dialog box.

Note: The port you use to monitor a Citrix MetaFrame server through a firewall depends on the configuration of your server.

Before setting up the Citrix MetaFrame Server monitor:

1. Make sure that Citrix MetaFrame Server has been installed and is running on a computer. If the computer running Citrix MetaFrame Server is running Windows 2000, make sure that the Remote Registry service is running on it.
2. Make sure that the computer on which you are running LoadRunner Tuning Module has administrator privileges on the machine running Citrix.
3. From the Console machine, map a network drive to the Citrix server machine. This ensures that the required authentication is provided to the Console to access the resource counters.
4. Launch PerfMon from the Console machine to enable the counters on the Citrix server. This allows you to monitor the same counters for the ICA Session object on the Citrix monitor.
5. In the System Topology window, add an Application Deployment element representing the server running Citrix. Assign the following values to the element's settings:
   - **O/S:** The version of Windows running on the Citrix machine
   - **Product**: Citrix MetaFrame
   - **Version:** XP

6 You can configure the Citrix monitor to view ICA Session object counters only if at least one session is being run on the Citrix server. If no "real" user has opened a connection with the Citrix server, you need to first initialize or run a Citrix Vuser against the server, and only then configure the Citrix Monitor and add the ICA Session counters. If you configure the Citrix monitor without first initializing or running a Citrix Vuser (or connecting to the Citrix server as a "real" user), you will not be able to view the ICA Session object.

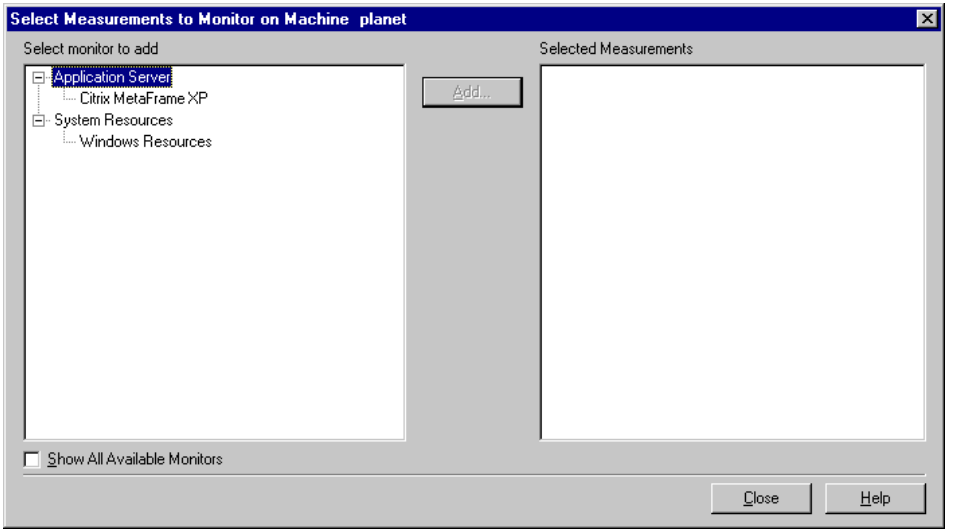
Note: Measurements that monitor instances are valid for the currently running Citrix session only. If you run this session step again, you will need to reconfigure the measurements that are instance-oriented.

Note: In order to monitor the different instances, ensure that the server login and logout procedures are recorded in the Vuser_init and Vuser_end sections respectively, and not in the Action section of the script. For more information, refer to the *Creating Vuser Scripts* guide.

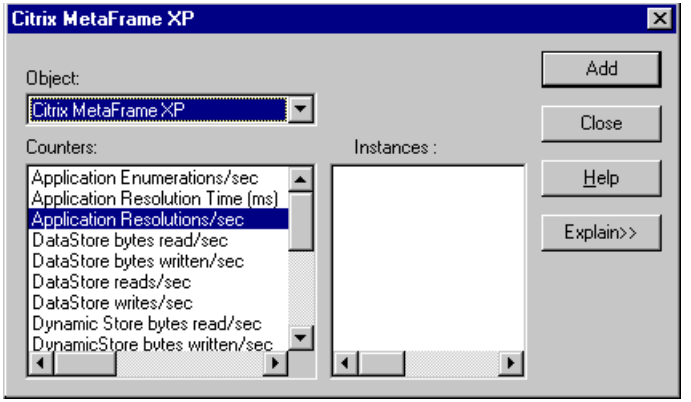
To configure the Citrix MetaFrame Server monitor:

1 Click **Monitors** to open the Monitors Configuration dialog box.

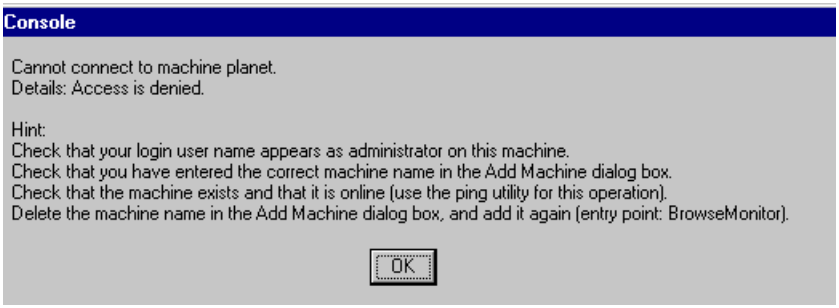
2 From the Server list, choose the server running Citrix and click **Add Monitor**. The Select Measurements to Monitor dialog box appears.



3 In the left section of the dialog box, click **Citrix MetaFrame XP** (in the Application Deployment category) and then click **Add**. The Citrix MetaFrame XP dialog box is displayed.



Note: If, instead of the Citrix MetaFrame XP dialog box, LoadRunner Tuning Module displays the following error message,



it means that the computer on which you are running LoadRunner Tuning Module does not have administrator privileges on the Citrix machine. To remedy this, close the LoadRunner Tuning Module session, acquire administrator privileges on the Citrix machine, and reopen the LoadRunner Tuning Module session. Then repeat this procedure, starting again with step 1 above.

Note: If the dialog box freezes after clicking Add, you may need to rebuild the localhost cache on the Citrix server machine.
For more information, refer to Documents IDs CTX003648 and CTX759510 in the Citrix Knowledge Base (http://knowledgebase.citrix.com/cgi-bin/webcgi.exe?New,KB=CitrixKB).

---

4 The Object listbox allows you to display three types of counters: Citrix MetaFrame XP, Citrix IMA Networking, and ICA Session. Choose the object whose counters you want to display. LoadRunner Tuning Module displays the object's counters in the Counters pane.

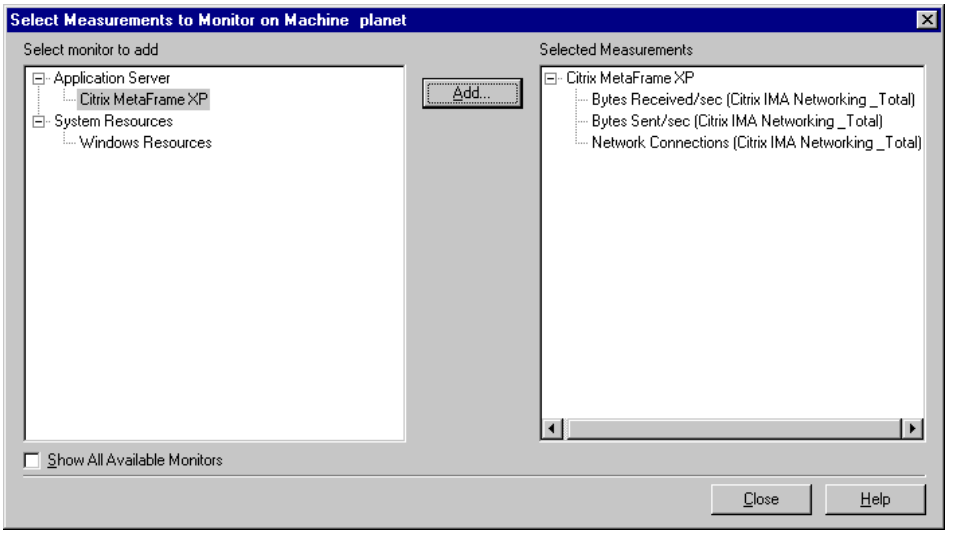
5 Select a counter and instance. You can select multiple counters using the **Ctrl** key. The instance is relevant only if multiple instances of the

highlighted counter are running. For a description of each counter, click **Explain>>** to expand the dialog box.

6 Click **Add** to add the counter to the list of counters that you want to monitor.

7 Repeat steps 5 through 6 until you've added all the Citrix counters that you want to monitor.

8 Click **Close**. The counters you selected are displayed in the Selected Measurements pane of the Select Measurements to Monitor dialog box.



9 Click **Close**.

Note: For troubleshooting tips and limitations, see the section on "Troubleshooting Server Resource Monitors," on page 645.

Citrix MetaFrame Performance Counters

The following table describes some of the counters that can be measured.

Non-Virtual Counters

Measurement	Description
% Disk Time	The percentage of elapsed time that the selected disk drive is busy servicing read or write requests.
% Processor Time	The percentage of time that the processor is executing a non-Idle thread. This counter is a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the Idle process in each sample interval, and subtracting that value from 100%. (Each processor has an Idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.
File data Operations/sec	The rate that the computer is issuing Read and Write operations to file system devices. It does not include File Control Operations.
Interrupts/sec	The average number of hardware interrupts the processor is receiving and servicing in each second. It does not include DPCs, which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Measurement	Description
Output Session Line Speed	This value represents the line speed from server to client for a session in bps.
Input Session Line Speed	This value represents the line speed from client to server for a session in bps.
Page Faults/sec	A count of the Page Faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in main memory. A Page Fault will not cause the page to be fetched from disk if that page is on the standby list, and hence already in main memory, or if it is in use by another process with whom the page is shared.
Pages/sec	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system Cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
Pool Nonpaged Bytes	The number of bytes in the Nonpaged Pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged Pool pages cannot be paged out to the paging file, but instead remain in main memory as long as they are allocated.
Private Bytes	The current number of bytes this process has allocated that cannot be shared with other processes.

Measurement	Description
Processor Queue Length	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Threads	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.
Latency – Session Average	This value represents the average client latency over the life of a session.
Latency – Last Recorded	This value represents the last recorded latency measurement for this session.
Latency – Session Deviation	This value represents the difference between the minimum and maximum measured values for a session.
Input Session Bandwidth	This value represents the bandwidth from client to server traffic for a session in bps.
Input Session Compression	This value represents the compression ratio for client to server traffic for a session.
Output Session Bandwidth	This value represents the bandwidth from server to client traffic for a session in bps.
Output Session Compression	This value represents the compression ratio for server to client traffic for a session.
Output Session Linespeed	This value represents the line speed from server to client for a session in bps.

Virtual Channel Counters

Measurement	Description
Input Audio Bandwidth	This value represents the bandwidth from client to server traffic on the audio mapping channel. This is measured in bps.
Input Clipboard Bandwidth	This value represents the bandwidth from client to server traffic on the clipboard mapping channel. This is measured in bps.
Input COM1 Bandwidth	This value represents the bandwidth from client to server traffic on the COM1 channel. This is measured in bps.
Input COM2 Bandwidth	This value represents the bandwidth from client to server traffic on the COM2 channel. This is measured in bps.
Input COM Bandwidth	This value represents the bandwidth from client to server traffic on the COM channel. This is measured in bps.
Input Control Channel Bandwidth	This value represents the bandwidth from client to server traffic on the ICA control channel. This is measured in bps.
Input Drive Bandwidth	This value represents the bandwidth from client to server traffic on the client drive mapping channel. This is measured in bps.
Input Font Data Bandwidth	This value represents the bandwidth from client to server traffic on the local text echo font and keyboard layout channel. This is measured in bps.
Input Licensing Bandwidth	This value represents the bandwidth from server to client traffic on the licensing channel. This is measured in bps.
Input LPT1 Bandwidth	This value represents the bandwidth from client to server traffic on the LPT1 channel. This is measured in bps.

Measurement	Description
Input LPT2 Bandwidth	This value represents the bandwidth from client to server traffic on the LPT2 channel. This is measured in bps.
Input Management Bandwidth	This value represents the bandwidth from client to server traffic on the client management channel. This is measured in bps.
Input PN Bandwidth	This value represents the bandwidth from client to server traffic on the Program Neighborhood channel. This is measured in bps.
Input Printer Bandwidth	This value represents the bandwidth from client to server traffic on the printer spooler channel. This is measured in bps.
Input Seamless Bandwidth	This value represents the bandwidth from client to server traffic on the Seamless channel. This is measured in bps.
Input Text Echo Bandwidth	This value represents the bandwidth from client to server traffic on the local text echo data channel. This is measured in bps.
Input Thinwire Bandwidth	This value represents the bandwidth from client to server traffic on the Thinwire (graphics) channel. This is measured in bps.
Input VideoFrame Bandwidth	This value represents the bandwidth from client to server traffic on the VideoFrame channel. This is measured in bps.
Output Audio Bandwidth	This value represents the bandwidth from server to client traffic on the audio mapping channel. This is measured in bps.
Output Clipboard Bandwidth	This value represents the bandwidth from server to client traffic on he clipboard mapping channel. This is measured in bps.
Output COM1 Bandwidth	This value represents the bandwidth from server to client traffic on the COM1 channel. This is measured in bps.

Measurement	Description
Output COM2 Bandwidth	This value represents the bandwidth from server to client traffic on the COM2 channel. This is measured in bps.
Output COM Bandwidth	This value represents the bandwidth from server to client traffic on the COM channel. This is measured in bps.
Output Control Channel Bandwidth	This value represents the bandwidth from server to client traffic on the ICA control channel. This is measured in bps.
Output Drive Bandwidth	This value represents the bandwidth from server to client traffic on the client drive channel. This is measured in bps.
Output Font Data Bandwidth	This value represents the bandwidth from server to client traffic on the local text echo font and keyboard layout channel. This is measured in bps.
Output Licensing Bandwidth	This value represents the bandwidth from server to client traffic on the licensing channel. This is measured in bps.
Output LPT1 Bandwidth	This value represents the bandwidth from server to client traffic on the LPT1 channel. This is measured in bps.
Output LPT2 Bandwidth	This value represents the bandwidth from server to client traffic on the LPT2 channel. This is measured in bps.
Output Management Bandwidth	This value represents the bandwidth from server to client traffic on the client management channel. This is measured in bps.
Output PN Bandwidth	This value represents the bandwidth from server to client traffic on the Program Neighborhood channel. This is measured in bps.
Output Printer Bandwidth	This value represents the bandwidth from server to client traffic on the printer spooler channel. This is measured in bps.

Measurement	Description
Output Seamless Bandwidth	This value represents the bandwidth from server to client traffic on the Seamless channel. This is measured in bps.
Output Text Echo Bandwidth	This value represents the bandwidth from server to client traffic on the local text echo data channel. This is measured in bps.
Output Thinwire Bandwidth	This value represents the bandwidth from server to client traffic on the Thinwire (graphics) channel. This is measured in bps.
Output VideoFrame Bandwidth	This value represents the bandwidth from server to client traffic on the VideoFrame channel. This is measured in bps.

40

Middleware Performance Monitoring

Using LoadRunner Tuning Module's Middleware Performance monitors, you can monitor the Tuxedo, PeopleSoft and IBM WebSphere MQ servers during a session step run and isolate server performance bottlenecks.

This chapter describes:

- Configuring the TUXEDO Monitor
- Configuring the IBM WebSphere MQ Monitor

About Middleware Performance Monitoring

A primary factor in a transaction's response time is the middleware performance usage. LoadRunner Tuning Module's Middleware Performance monitors provide you with information about the middleware performance usage of the Tuxedo and IBM WebSphere MQ servers during a session step execution. In order to obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the session step.

The IBM WebSphere MQ monitor is used to monitor channel and queue performance counters on an IBM WebSphere MQ (version 5.x) Server.

The Tuxedo monitor can monitor the server, load generator machine, workstation handler, and queue in a Tuxedo system. Note that in order to run the Tuxedo monitor, you must install the Tuxedo client libraries on the machine you want to monitor.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Configuring the TUXEDO Monitor

The Tuxedo monitor allows you to measure and view the performance of your Tuxedo application server.

Note: If Tuxedo 7.1 or higher is installed on the Console machine, more than one Tuxedo application server can be monitored at a time. However, if Tuxedo 6.5 or below is installed on the Console machine, only one Tuxedo application server can be monitored at a time.

Before you set up the monitor:

1. Ensure that a Tuxedo workstation client (not a native client) is installed on the Console machine. Use a Tuxedo 6.x client if a Tuxedo 6.x server is used, and Tuxedo 7.1 or above if a Tuxedo 7.1 or above server is used. If you use a Tuxedo 6.5 or earlier server, you can still use a Tuxedo 7.1 or later client in order to monitor it, provided that you set the WSINTOPPRE71 environment variable to "yes".

Note: A Tuxedo workstation client communicates with the application server over the network, and is not required to run the Tuxedo application server on the same machine. A native client can only communicate with the Tuxedo application server if it is part of the relevant Tuxedo domain.

---

2. Define the Tuxedo environment variables on the Console machine—set the TUXDIR variable to the Tuxedo installation directory (for example, V:\environ\32\Tuxedo8.0) , and add the Tuxedo bin directory to the PATH variable.
3. Ensure that the workstation listener (WSL) process is running. This enables the application server to accept requests from workstation clients. Note that the address and port number used to connect to the application server must match those dedicated to the WSL process.

Note: For information on configuring the WSL, refer to the BEA Tuxedo Web site (http://edocs.beasys.com/tuxedo/tux81/rf5/rf5101.htm#1534543).

To configure the Tuxedo monitor:

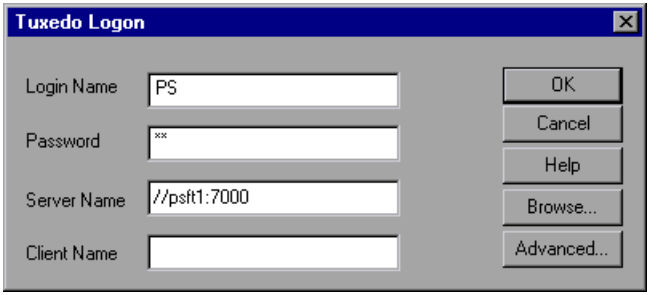
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Middleware Component category, select **Tuxedo**, and then click **Add**.

The Tuxedo Logon dialog box is displayed.



5 Enter the Tuxedo server Login Name, Password, Server Name and Client Name. If a Tuxedo server was previously monitored, its name is displayed in the Server Name box.

The format of the server name is //<machine name>:<port number>. Alternatively, you can specify the IP address or the hexadecimal format used by old versions of Tuxedo. Note that you cannot use quotation marks.

Note: Logon information is located in the Logon section of the **tpinit.ini** file in the recorded script's directory. It is recommended that you use the Browse button and select the **tpinit.ini** file from a recorded script, rather than enter the values manually. You can also determine the client name from the **lrt_tpinitialize** statement in the recorded script.

In the following example of a *tpinit.ini* file, the Tuxedo monitor was configured for a server named URANUS using port 65535, and a client named bankapp. The logon user name was Smith and the password was mypasswd.

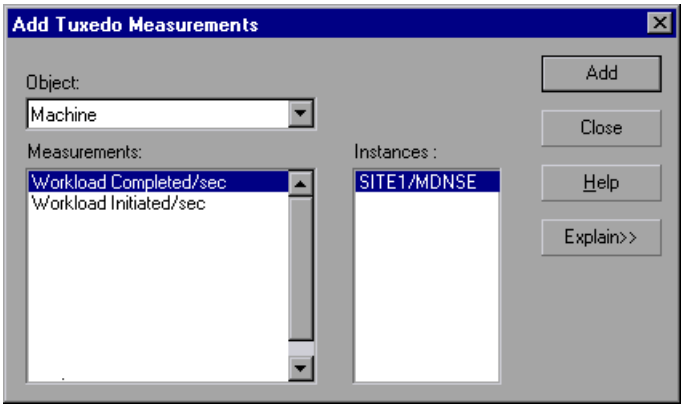
```
[Logon]
LogonServername=//URANUS:65535
LogonUsrName=Smith
LogonCltName=bankapp
LogonGrpName=
LogonPasswd=mypasswd
LogonData=
```

6 To authenticate the Tuxedo monitor, click **Advanced**, and enter the authentication data as a hexadecimal string (beginning with "0x") in the data box. The authentication data value can be obtained from the **tpinit.ini** file of an existing Tuxedo script.

Note: If you are using Tuxedo 6.5 or below, the monitor can only connect to one application server during a Console session. Once it connects to an application server, that server is the only one used by the monitor until the Console is closed. This applies even when all of the counters are deleted from the monitor.

---

7 Click **OK**. The Add Tuxedo Measurements dialog box opens.



8 Select a Tuxedo object from the Object list. Select the measurements and instances you want to monitor. For a list of the available performance counters, see page 618.

9 Click **Add** to place the selected object on the resource list. Add all the desired objects to the list, and click **Close**.

10 Click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Tuxedo Performance Counters

The following table describes the default counters that can be measured. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for relevant queues.

Monitor	Measurements
Machine	**% Busy Clients** - The per cent of active clients currently logged in to the Tuxedo application server which are waiting for a response from the application server.
	Active Clients - The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients - The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers - Number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions - Number of in use transaction table entries on this machine.
	Idle Clients - The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second - The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second - The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers** - The per cent of active servers currently handling Tuxedo requests.
	Active Servers - The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers - The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers - The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued - The total number of messages which have been placed on the queue.
Server	**Requests/second** - How many server requests were handled per second
	Workload/second -The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec** - The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec - The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec - The number of messages received by the workstation handler, per second.
	Messages Sent/sec - The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec - The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

Configuring the IBM WebSphere MQ Monitor

To use the IBM WebSphere MQ monitor you must first install the IBM WebSphere MQ client, configure the MQ server environment to monitor events, and then select the measurements you want to monitor using the IBM WebSphere MQ Add Measurements dialog box.

Note: LoadRunner Tuning Module IBM WebSphere MQ Monitor supports MQ servers on Windows platforms only.

Note: To monitor the MQ middleware performance monitor, the Windows user must be part of the Administration Group of the IBM WebSphere MQ server.

Connecting to the IBM WebSphere MQ Server

The IBM WebSphere MQ monitor connects to the IBM WebSphere MQ server (via the MQ Client Connection installed on the Console machine). In MQ Client environments, MQ does not run on the client machine. Instead, the client machine connects to an MQ Server instance, and uses the Server's resources as if they were local to the client machine.

Note: Note: The IBM WebSphere MQ monitor provides resource usage information for machines running the IBM MQ Server (version 5.2) for Windows monitoring.

Before you set up the monitor:

Ensure that an IBM WebSphere MQ Client Connection (version 5.21 only) is installed on the Console machine.

Note: For additional information on the IBM WebSphere MQ Server/Client, please refer to the IBM MQSeries Web site (http://www-3.ibm.com/software/ts/mqseries/library/manuals/index.htm).

Configuring the Server Environment to Monitor Events

The LoadRunner Tuning Module MQ Monitor retrieves event messages from two standard MQSeries queues only:

- SYSTEM.ADMIN.PERFM.EVENT – performance events, such as "queue depth high"
- SYSTEM.ADMIN.CHANNEL.EVENT – channel events, such as "channel stopped"

Events must be enabled for the queue manager (and in many cases, on the applicable object, as well). Performance events are enabled by setting attributes for the queue on the MQ Server. Channel events are enabled by default, and cannot be disabled.

Note: The IBM WebSphere MQ monitor does not retrieve data from a queue manager after the queue manager has been restarted.

To enable performance events for the Queue Manager:

1 Use the following MQSC command: ALTER QMGR PERFMEV(ENABLED).

2 Set the following attributes for the queue:

Measurement	Set Event Attributes
Event - Queue Depth High	• QDEPTHHI(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPHIEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.
Event - Queue Depth Low	To enable the event for a queue, the following attributes of the queue must be set: • QDEPTHLO(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPLOEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.
Event - Queue Full	• QDEPTHHI(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPMAXEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.

Measurement	Set Event Attributes
Event - Queue Service Interval High	• QSVCINT(integer) – where integer is a value expressed as milliseconds, in the range of 0 and 999,999,999, inclusive. Note: this value is shared with Queue Service Interval OK. • QSVCIEV(type) – where type is the word "HIGH", "OK", or "NONE", enabling service interval high events, enabling service interval ok events, or disabling the generation of the event, respectively.
Event - Queue Service Interval OK	• QSVCINT(integer) – where integer is a value expressed as milliseconds, in the range of 0 and 999,999,999, inclusive. Note: this value is shared with Queue Service Interval High. • QSVCIEV(type) – where type is the word "HIGH", "OK", or "NONE", enabling service interval high events, enabling service interval ok events, or disabling the generation of the event, respectively.

Note: If you encounter an MQ Server error message (starting with the characters MQRC_), please refer to the Reason Codes section of the IBM MQSeries Web site (http://www-3.ibm.com/software/ts/mqseries/library/manuals/mqw20/AMQ43M32.HTM#HDRMQSCRN).

Configuring the IBM WebSphere MQ Monitor in the Console

After you have installed the MQ Client on the Console, and configured the server environment to monitor events, you can specify which resources you want to measure.

To configure the IBM WebSphere MQ monitor:

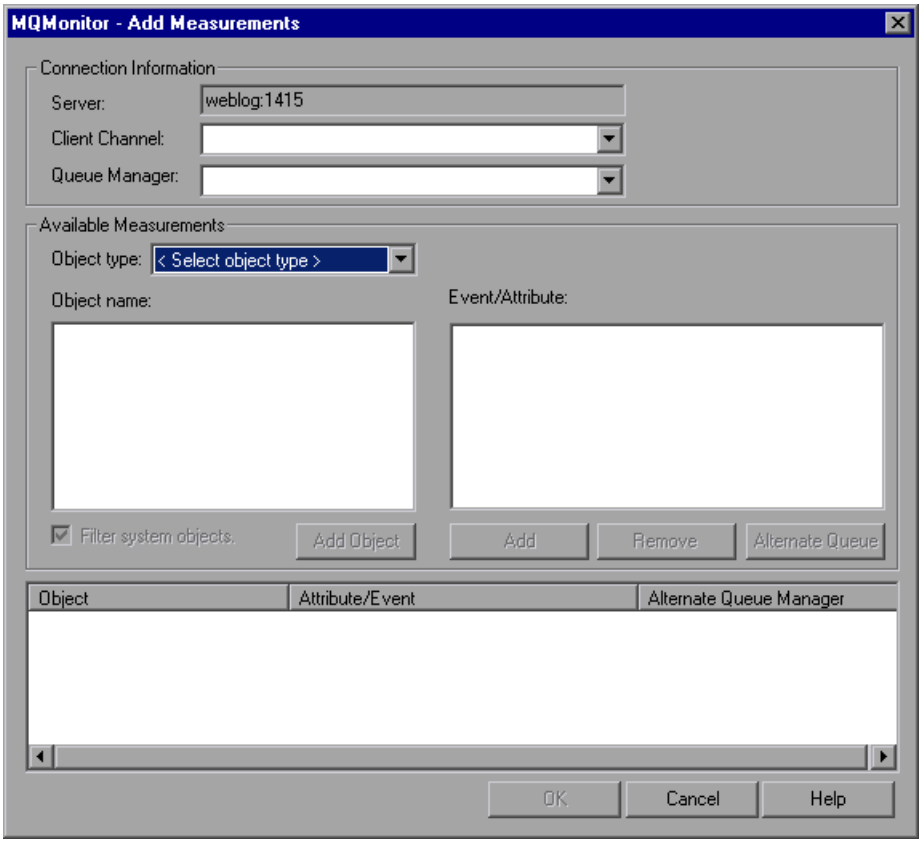
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Middleware Component category, select **IBM WebSphere MQ**, and then click **Add**.

The IBM WebSphere MQ Add Measurements dialog box opens.



5 In the Connection Information section, enter the name of the channel through which a client connection is made to an MQ Server in the Client Channel box.

You can set up a specific channel on an MQ Server instance, or use the default "SYSTEM.DEF.SVRCONN" channel. If the client channel is undefined, the MQ Server will be inaccessible via client connections (the MQ Monitor will not work, as it will not be able to connect to the queue manager which it is supposed to monitor).

Note: User entries for any text box are limited to 48 characters.

---

6 Enter the name of the queue manager to be monitored in the Queue Manager box.

The monitor is not restricted to monitoring only the queue manager to which it is connected. You can configure multiple queue managers to write to the event queue of a central queue manager for centralized monitoring (this applies to Events only, not polled object attributes). All events contain a queue manager attribute identifying their source.

Note: A queue manager can only be accessed by one Console or monitoring application at any one time.

---

7 In the Available Measurements section, select an object type.

A list of previously added objects of the selected object type appear in the Object name list. A list of attributes or events applicable to the selected object type appear in the Events/Attributes list.

The names of monitored objects, event/attribute selected, and alternate queue managers, are listed in the monitored objects pane.

8 By default, only user-defined objects are displayed in the Object name list. To show all objects, clear the **Filter System Objects** check box. You can modify the filter settings, in the *<LoadRunner Tuning Module_installation>\dat\monitors\mqseries.cfg* file.

9 Select an object or add a new object to the Object name list. To add a new object name, click the **Add Object** button. In the Add Object Name dialog box, enter the name of an object to be monitored and click **OK**. The dialog box closes and the name of the object appears in the Object name list.

10 Select the attributes or events to be measured from the Attribute/Event box. The list of attributes or events is applicable to the selected object type.

For a list of the available performance counters, see page 627.

Note: To enable the event for a queue, ensure that the attributes for the queue have been set. For more information, refer to "Configuring the Server Environment to Monitor Events," on page 621.

---

11 If the event configured for monitoring is from a remote queue manager (other than the one identified in the queue manager field of the IBM WebSphere MQ Add Measurements dialog box), click the **Alternate Queue** button. Enter the name of an alternate queue manager in the Alternate Queue dialog box, and click **OK.**

Note: When you add an alternate queue manager, this becomes the default queue manager for any events that you subsequently add. To return to the queue manager to which you are connected, enter that name in the Alternate Queue Manager dialog box.

---

12 Click **Add** to add the object measurements to the monitored objects list. The name of object, it's events and attributes, and queue managers, are listed in the monitored objects pane.

13 To remove a monitored object event or attribute, select the object measurement in the monitored objects pane, and click **Remove**. The entry is deleted from the monitored objects list.

14 Add all the desired counters to the monitored objects list, and click **OK**.

15 Click **OK** in the Select Measurements to Monitor dialog box to activate the

monitor.

IBM WebSphere MQ Performance Counters

The following tables list the available IBM WebSphere MQ monitor measurements:

Queue Performance Counters

Measurement	Description
Event - Queue Depth High (events per second)	An event triggered when the queue depth reaches the configured maximum depth.
Event - Queue Depth Low (events per second)	An event triggered when the queue depth reaches the configured minimum depth.
Event - Queue Full (events per second)	An event triggered when an attempt is made to put a message on a queue that is full.
Event - Queue Service Interval High (events per second)	An event triggered when no messages are put to or retrieved from a queue within the timeout threshold.
Event - Queue Service Interval OK (events per second)	An event triggered when a message has been put to or retrieved from a queue within the timeout threshold.
Status - Current Depth	Current count of messages on a local queue. This measurement applies only to local queues of the monitored queue manager.
Status - Open Input Count	Current count of open input handles. Input handles are opened so that an application may "put" messages to a queue.
Status - Open Output Count	Current count of open output handles. Output handles are opened so that an application may "get" messages from a queue.

Channel Performance Counters

Measurement	Description
Event - Channel Activated (events per second)	Event generated when a channel, waiting to become active but inhibited from doing so due to a shortage of queue manager channel slots, becomes active due to the sudden availability of a channel slot.
Event - Channel Not Activated (events per second)	Event generated when a channel, attempts to become active but inhibited from doing so due to a shortage of queue manager channel slots.
Event - Channel Started (events per second)	Event generated when a channel is started.
Event - Channel Stopped (events per second)	Event generated when a channel is stopped, regardless of source of stoppage.
Event - Channel Stopped by User (events per second)	Event generated when a channel is stopped by a user.
Status - Channel State	The current state of a channel. Channels pass through several states from STOPPED (inactive state) to RUNNING (fully active state). Channel states range from 0 (STOPPED) to 6 (RUNNING).
Status - Messages Transferred	The count of messages that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Buffer Received	The count of buffers that have been received over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.

Measurement	Description
Status - Buffer Sent	The count of buffers that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Received	The count of bytes that have been received over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Sent	The count of bytes that have been sent over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.

41

Application Traffic Management Monitoring

Using LoadRunner Tuning Module's Application Traffic Management monitor, you can monitor the BIG-IP load balancing device during a session step run and isolate server performance bottlenecks.

This chapter describes:

➤ Configuring the F5 BIG-IP Monitor

About Application Traffic Management Monitoring

LoadRunner Tuning Module's F5 BIG-IP monitor provides you with information about the content of event logs and other data from F5 BIG-IP load balancing device using SNMP during a session step execution. In order to obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the session step.

Configuring the F5 BIG-IP Monitor

To use the F5 BIG-IP monitor, you must first configure the F5 BIG-IP monitor on the SiteScope machine, and then select the counters you want the F5 BIG-IP monitor to measure. You select these counters using the Console's F5 BIG-IP dialog box.

Before setting up the F5 BIG-IP monitor:

1. Make sure that SiteScope has been installed on a server. SiteScope is the application that is used to monitor the F5 BIG-IP server. Although you can install SiteScope on the Console machine, we recommend installing it on a dedicated server.
2. On the machine where SiteScope is installed, configure SiteScope to monitor the required F5 BIG-IP machines. For more information on configuring the SiteScope server, refer to the SiteScope User Guide (http://www.freshwater.com/SiteScope/UGtoc.htm).

Note: When you assign a name to a monitor, include the server name in the monitor name. This avoids any confusion as to which host the monitor belongs.

---

3. Verify that SiteScope is collecting the required data from the servers it is monitoring. From the SiteScope Panel, select the monitor group polling the F5 BIG-IP server machines, and check that the monitor displays a list of server measurements in the Status column.

Configuring the F5 BIG-IP monitor on the SiteScope machine:

1. Open the SiteScope Add Monitors to Group page, and click **F5 BIG-IP Monitor**.
2. Click **Choose Server**.
3. In the server field, enter the name or IP address of the F5 BIG-IP server that you want to monitor.
4. Enter the community for the SNMP object. The default community is public.

5 Enter the appropriate value in the Retry Delay field. The default value is 1 second.

6 Enter the appropriate value in the Timeout field. The default value is 5 seconds.

7 Click **Browse Counters** and select your desired counters.

8 Click **Choose Counters**, and then click **Add Monitor**.

Configuring the F5 BIG-IP monitor on the Console machine:

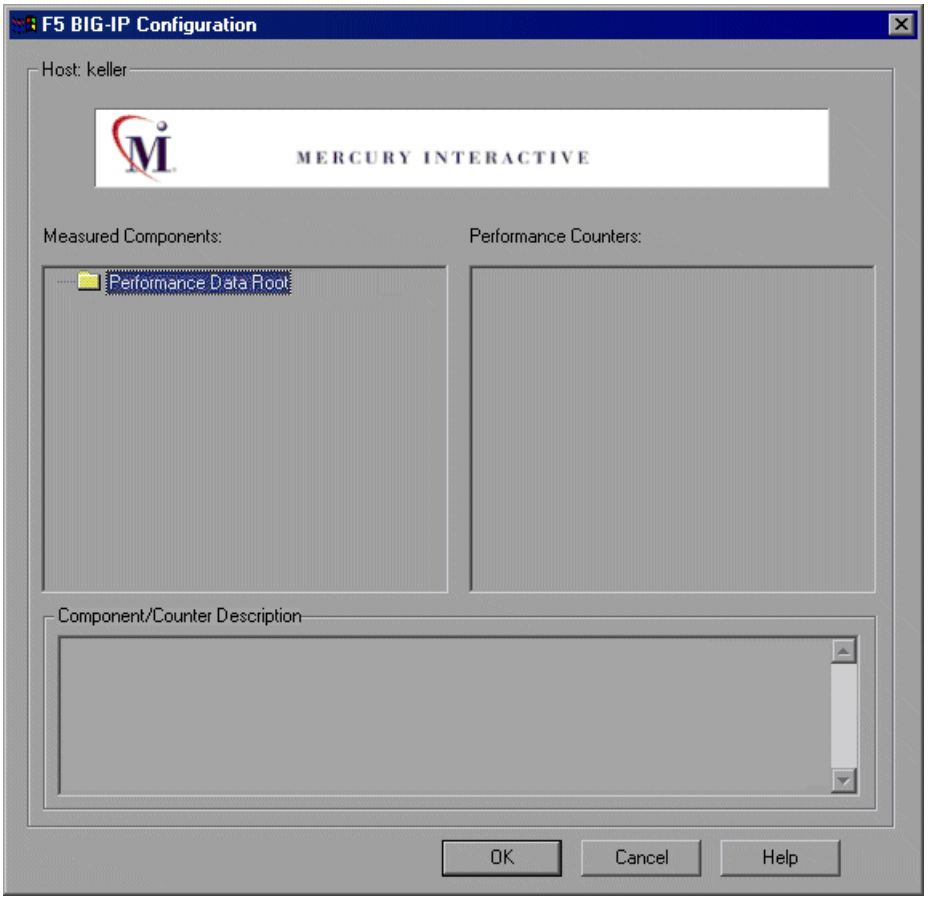
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running F5 BIG-IP.

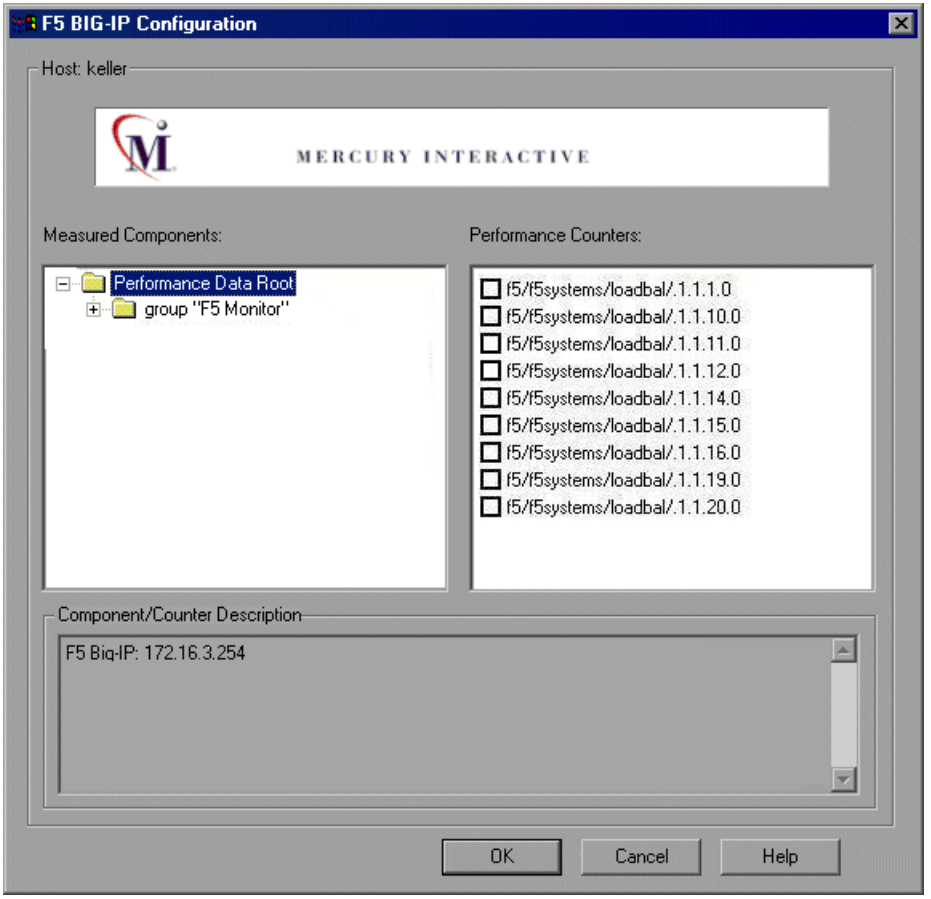
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left pane, expand the Load Balancer category, click **F5 BIG-IP**, and click **Add**.

The F5 BIG-IP Monitor Configuration dialog box is displayed.



5 In the Measured Components pane, locate the F5 BIG-IP measurement that you are monitoring and click it. The performance counters that F5 BIG-IP is monitoring on the selected component are displayed in the Performance Counters pane.



The following table shows the default counters that can be measured:

Measurement
pktsin
pksout
concur
portdeny
uptime
droppedin
droppedout
MemoryUsed

6 Check the required performance counters in the Performance Counters pane.

7 When you have selected the performance counters for the F5 BIG-IP measurements you are monitoring, click **OK** to close the F5 BIG-IP Configuration dialog box. The **Select Measurements to Monitor** dialog box appears with the selected F5 BIG-IP measurements in the **Selected Measurements** pane.

8 Click **OK** in the **Select Measurements to Monitor** dialog box, and click **OK** in the Monitors Configuration dialog box, to activate the F5 BIG-IP monitor.

Note: For troubleshooting tips and limitations, see "Troubleshooting Server Resource Monitors," on page 645.

42

Security Monitoring

When you run certain security scripts, you can use LoadRunner Tuning Module's security graphs to view information about the simulated attacks on the server.

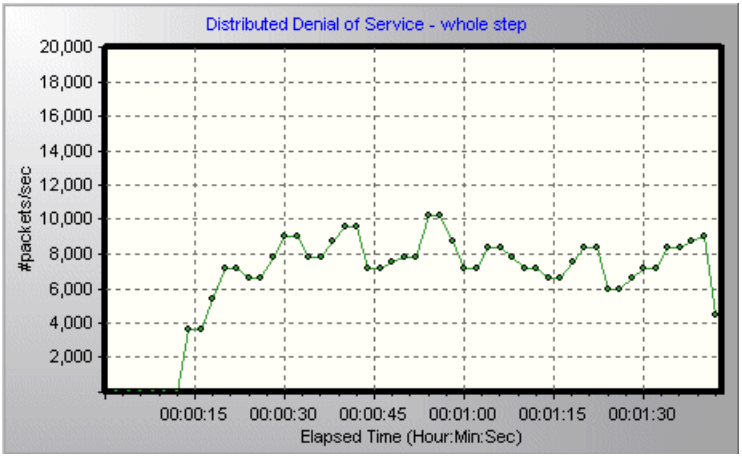
This chapter describes:

➤ Distributed Denial of Service Graph

Distributed Denial of Service Graph

The Distributed Denial of Service graph displays the number of packets per second sent to the specified target to cause denial of service.

The x-axis represents the elapsed time. The y-axis represents the number of packets sent per second.



Note: To obtain data for this graph, you need to execute one of the following canned security scripts: SYN FLOOD DDOS Attack or UDP Echo DDOS Attack.

43

SiteScope Server Monitoring

You can monitor a machine's system resource usage during a session step run using LoadRunner Tuning Module's SiteScope Server monitor.

This chapter describes:

➤ Configuring the SiteScope Monitor

About SiteScope Server Monitoring

Using the LoadRunner Tuning Module SiteScope monitor, you can monitor the SiteScope resources on a machine during a session step run, and determine why a bottleneck occurred on a particular machine.

The SiteScope monitor can measure server, network, and processor performance counters. For detailed information on the performance counters that SiteScope can monitor, refer to the relevant SiteScope documentation.

To obtain data for this graph, you must specify the machine you want to monitor and which resources to monitor for each machine. You can also add or remove machines and resources during the session step run.

Configuring the SiteScope Monitor

You select measurements to poll from SiteScope using the SiteScope Monitor Configuration dialog box.

Before setting up the SiteScope monitor:

1. Make sure that SiteScope has been installed on a machine. If SiteScope is not installed on the same machine as the Console, verify that the SiteScope machine is accessible from the Console machine.
2. On the machine where SiteScope is installed, configure SiteScope to monitor the required machines. When you assign a name to a monitor, include the hostname in the monitor name. This avoids confusion about the host to which the machine belongs.

Note: The monitor title cannot include "\", (for example, "CPU Utilization on \\Superman".

Note: SiteScope's default sampling rate is 10 minutes, and its minimum rate 15 seconds.

---

3. Verify that SiteScope is collecting the required data from the machines it is monitoring.
4. In the System Topology window, add a SiteScope Server element for the machine on which SiteScope is running.

You can configure the SiteScope monitor in one of the following ways:

- In the System Topology window
- In the Session tab

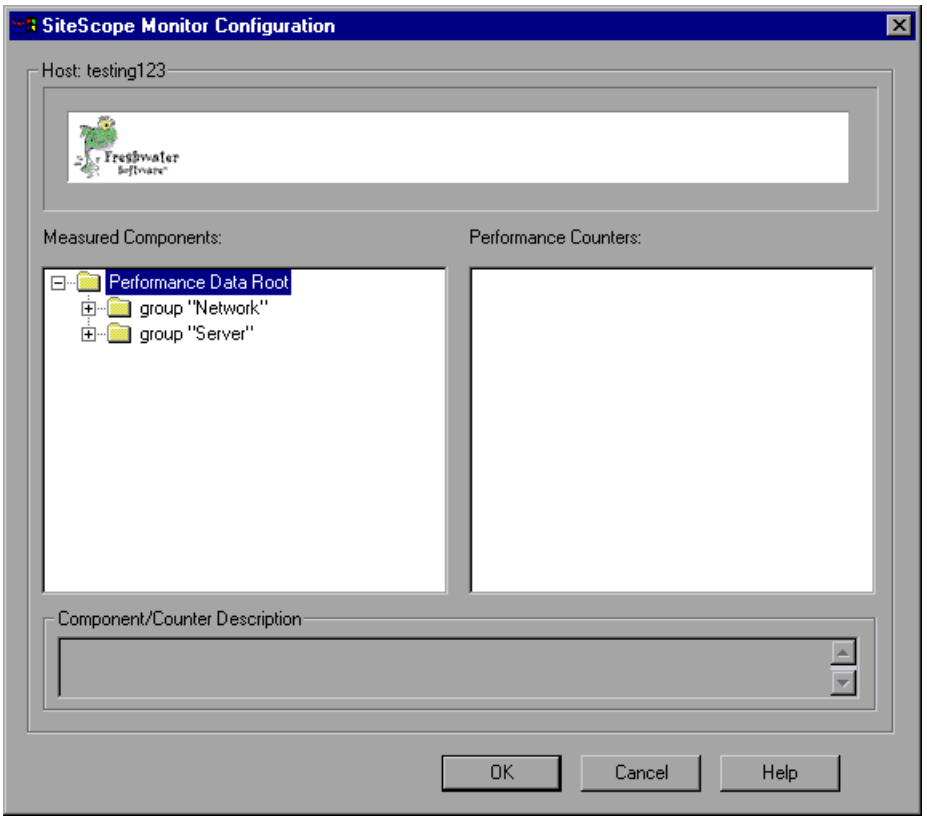
To configure the SiteScope monitor via the System Topology window:

1 Click the SiteScope Server element to select it.

2 In the **Element Monitors** tab, click **Add**. The **Select Measurements to Monitor** dialog box appears.

3 Check the Show All Available Monitors box, expand the SiteScope Server element, and then click SiteScope.

4 Click **Add**. The SiteScope Monitor Configuration dialog box is displayed.

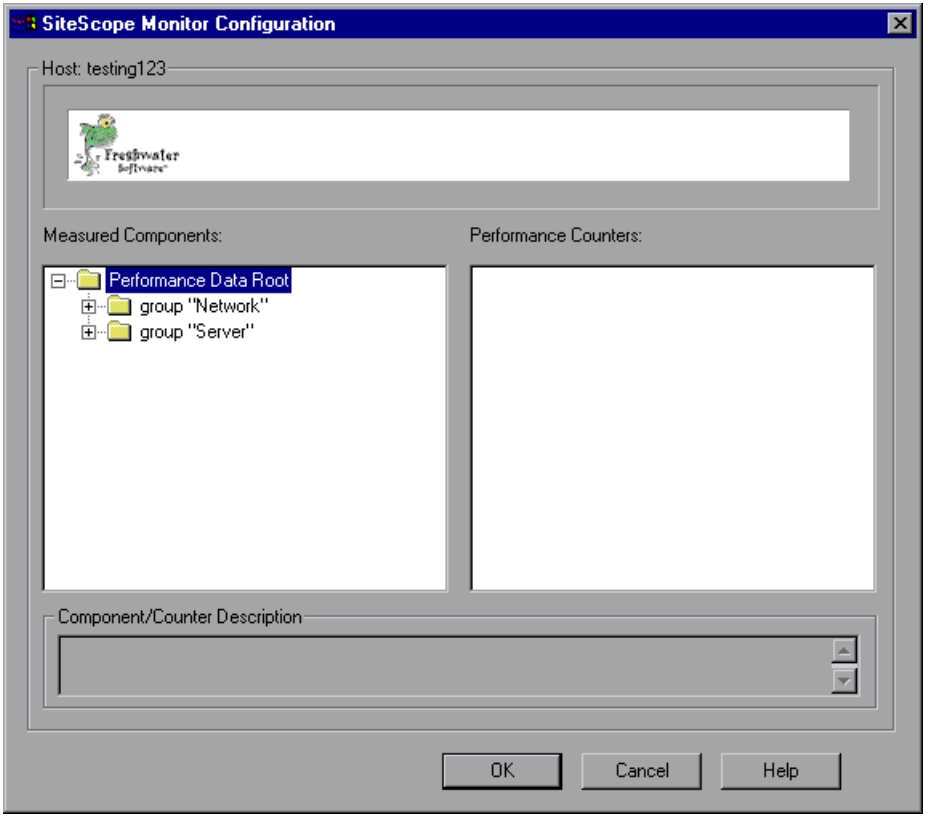


5 Continue with step 3 in the following section.

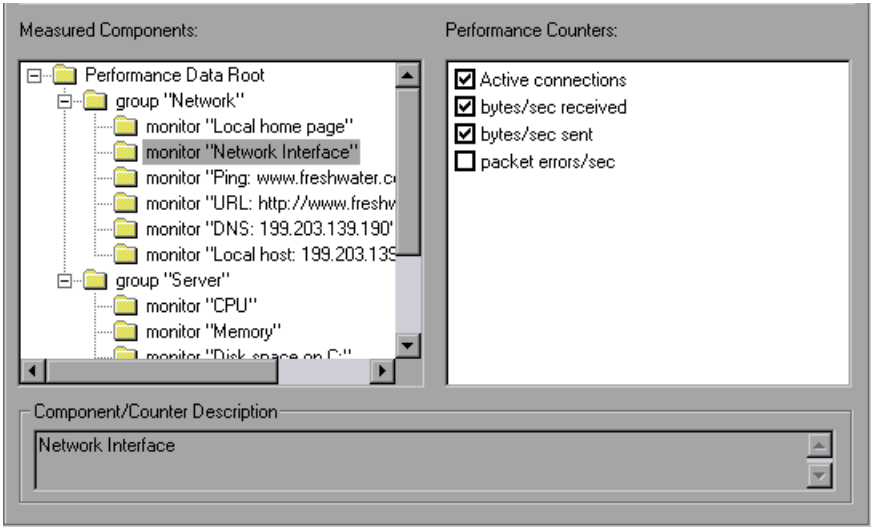
To configure the SiteScope monitor via the Session tab:

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running SiteScope and click **Add**. The SiteScope Monitor Configuration dialog box is displayed.



3 In the Measured Components pane, locate the SiteScope measurement that you are monitoring and click it. The performance counters that SiteScope is monitoring on the selected component are displayed in the Performance Counters pane.



4 Check the required performance counters in the Performance Counters pane and click **OK**.

5 The **Select Measurements to Monitor** dialog box appears with the selected SiteScope measurements in the Selected Measurements pane. Click **Close** in the Select Measurements to Monitor dialog box.

Note: If you are activating the SiteScope monitor from the Session tab, click **OK** in the Monitors Configuration dialog box to activate the monitor.

44

Troubleshooting Online Monitors

LoadRunner Tuning Module monitors allow you to view the performance of the session step during execution.

The following sections describe several tips and known issues relating to the online monitors.

- Troubleshooting Server Resource Monitors
- Troubleshooting the Network Delay Monitor
- Network Considerations

Troubleshooting Server Resource Monitors

In order to monitor resources on a server machine, you must be able to connect to that machine. If monitoring is unsuccessful and LoadRunner Tuning Module cannot locate the specified server, make sure that the specified server is available. Perform a "ping" operation by typing ping <*server_name*> from the Console machine command line.

Once you verify that the machine is accessible, check this table for additional tips on troubleshooting the monitor.

Problem	Solution
Cannot monitor a Windows machine on a different domain, or "access denied."	To gain administrative privileges to the remote machine, perform the following from the command prompt: net use \\<*MachineName*>/ user:[<*Domain*>\<*RemoteMachineUsername*>] At the password prompt, enter the password for the remote machine.

Problem	Solution
Cannot monitor an NT/Win 2000 machine (An error message is issued: "computer_name not found" or "Cannot connect to the host")	The NT/Win 2000 machine you want to monitor only enables monitoring for users with administrator privileges. In order to allow monitoring for non-admin users, you must grant read permission to certain files and registry entries (Microsoft tech-note number Q158438.) The required steps are: **a.** Using Explorer or File Manager, give the user READ access to: %windir%\system32\PERFCxxx.DAT %windir%\system32\PERFHxxx.DAT where *xxx* is the basic language ID for the system—for example, 009 for English. These files may be missing or corrupt. If you suspect this; expand these files off of the installation cd. **b.** Using REGEDT32, give the user READ access to: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib and all sub keys of that key. **c.** Using REGEDT32, give the user at least READ access to: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg
Some Win 2000 counters cannot be monitored from an NT machine.	Run the Console on a Win 2000 machine.
Some Windows default counters are generating errors	Remove the problematic counters and add the appropriate ones using the "Add Measurement" dialog box.
You cannot get performance counters for the SQL server (version 6.5) on the monitored machine.	There is a bug in SQL server version 6.5. As a workaround, give read permission to the following registry key at the monitored machine (use regedt32): HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer (Microsoft tech-note number Q170394)

Problem	Solution
The selected measurements are not displayed in the graph.	Ensure that the display file and online.exe are registered. To register the monitor dll's, without performing a full installation, run the *set_mon.bat* batch file located in *<LoadRunner Tuning Module root folder>/bin*.
When monitoring a Windows machine, no measurements appear in the graph.	Check the built-in Windows Performance Monitor. If it is not functional, there may be a problem with the communication setup.
When monitoring a UNIX machine, no measurements appear in the graph.	Ensure that an *rstatd* is running on the UNIX machine (Refer to Chapter 29, "System Resource Monitoring.").
Cannot monitor one of the following Web servers: MS IIS, MS ASP, or ColdFusion	Refer to problem above, "Cannot monitor a Windows machine."
Cannot monitor the WebLogic (JMX) server	Open the *<LoadRunner Tuning Module root folder>\dat\monitors\WebLogicMon.ini* file, and search for: [WebLogicMonitor] JVM=javaw.exe Change javaw.exe to java.exe. A window containing trace information opens.

Troubleshooting the Network Delay Monitor

If monitoring is unsuccessful and LoadRunner Tuning Module cannot locate the source or destination machines, make sure that the specified machines are available to your machine. Perform a "ping" operation. At the command line prompt, type:

```
ping server_name
```

To check the entire network path, use the trace route utility to verify that the path is valid.

For Windows, type tracert <*server_name*>.

For UNIX, type traceroute <*server_name*>.

If the monitoring problem persists once you verify that the machines are accessible and that the network path is valid, perform the following procedures:

1) If you are using the TCP protocol, run *<LoadRunner Tuning Module root folder\bin\webtrace.exe* from the source machine to determine whether the problem is related to the Console, or the WebTrace technology on which the Network Delay monitor is based. If you are using the UDP or ICMP protocols, the problem must be related to the Console and not WebTrace, since these protocols are not WebTrace technology-based.

2) If you receive results by running *webtrace.exe*, the problem is related to the Console. Verify that the source machine is not a UNIX machine, and contact Mercury Interactive's Customer Support with the following information:

- the Console log file, *drv_log.txt*, located in the temp directory of the Console machine.
- the *traceroute_server* log file, located on the source machine.
- the debug information located in the *TRS_debug.txt* and *WT_debug.txt* files in the path directory. These files are generated by adding the following line to the [monitors_server] section of the *<LoadRunner Tuning Module root folder\dat\mdrv.dat* file, and rerunning the Network monitor:

```
ExtCmdLine=-traceroute_debug path
```

3) If you do not receive results by running *webtrace.exe*, the problem is related to the WebTrace technology, on which the Network Delay monitor is based. Perform the following procedures on the source machine:

- Verify that the *packet.sys* file (the Webtrace driver) exists in the WINNT\system32\drivers directory.
- Check whether a driver (such as "Cloud" or "Sniffer") is installed on top of the network card driver. If so, remove it and run WebTrace again.
- Verify that there are administrator permissions on the machine.
- Using ipconfig /all, check that only one IP address is assigned to the network card. WebTrace does not know how to handle multiple IP addresses assigned to the same card (IP spoofing).
- Check the number of network cards installed. Run webtrace –devlist to receive a list of the available network cards.
- If there is more than one card on the list, run webtrace -dev <dev_name> <destination>, where <dev_name> is one of the network card names shown in the list. If you discover that WebTrace is binding to the wrong card, you can use webtrace set_device <dev_name> in order to set a registry key that instructs WebTrace to use a specified card instead of the default one.
- Verify that the network card is of the Ethernet type.
- Contact Mercury Interactive's Customer Support with the output of webtrace.exe –debug (for example, webtrace.exe –debug www.mercint.com) and ipconfig /all on the machine.

Network Considerations

If you notice extraordinary delays on the network, refer to one of the following sections to increase the performance:

- Network Bandwidth Utilization
- Ethernet-bus Based Networks

Network Bandwidth Utilization

In most load-testing session steps, the network card has little impact on session step performance. Network cards are manufactured to handle the bandwidth of the physical network layer. Packets are transferred over an Ethernet at a rate that complies with IEEE 803.x standards. If the network becomes a bottleneck, the issue is not the brand of the network card, but rather the bandwidth limitations on the physical layer (--i.e. Ethernet, FDDI, ATM, Ethernet Token-ring, etc.).

That is, instead of load testing over a T10 line, upgrade your line to DS3 (45Mbps), or T100 (100Mbps).

Below are a few tips that will help qualify the need to upgrade the network:

1) Run the performance monitor on the Vuser load generators. As the number of Vusers increases, check the network byte transfer rate for saturation. If a saturation point has been reached, do not run any more Vusers without upgrading the network—otherwise performance of Vusers will degrade. Degradation is exponential in networking environments.

2) Run the performance monitor on the server machine. Run many Vusers on several load generator machines. Check the kernel usage and network transfer rate for saturation. If saturation is reached with less than the desired Vuser load, upgrade the network.

3) Every network has a different Maximum Transmission Unit or MTU, which is set by the network administrator. The MTU is the largest physical packet size (in bytes) that a network can transmit. If a message is larger than the MTU, it is divided into smaller packets before being sent.

If clients and servers are passing large data sets back and forth, instruct the network administrator to increase the MTU in order to yield better

bandwidth utilization. Ideally, you want the MTU to be the same as the smallest MTU of all the networks between your machine and a message's final destination.

If you send a message that is larger than one of the MTUs, it will be broken up into fragments, slowing transmission speeds. If the MTU is too high, it may cause unintended degradation. Trial and error is the only sure way of finding the optimal MTU, but there are some guidelines that can help. For example, most Ethernet networks have an MTU of 1500.

If the desired MTU reduces performance, upgrade the network or reduce the MTU to improve performance.

Ethernet-bus Based Networks

The following guidelines apply to Ethernet-bus based networks:

Networks with only 2 active machines communicating yield a maximum of 90% bandwidth utilization.

Networks with 3 active machines communicating yield a maximum of approximately 85% bandwidth utilization.

As the number of active machines on the network increases, the total bandwidth utilization decreases.

Part VI

Tuning Your System

45

Tuning Your System from the Console

Once you've run session steps on your topology and analyzed the test results, you use the Console to administer and tune your hosts and services from a remote location. You continue this process until your system reaches optimal performance.

This chapter includes the following topics:

- Supported Operating Systems
- Applications That LoadRunner Tuning Module Can Tune
- Tuning Flow
- Host Requirements
- Tuning Through a Firewall
- Configuring Host Connection Parameters
- Connecting to the Host Computer
- Remote Installation of a Tuning Agent
- Local Installation of a Tuning Agent
- Viewing the Host Information
- Viewing Windows Services
- Using Expert Mode
- Viewing Differences Only
- Comparing Hosts and Services
- Viewing and Changing Tuning Parameter Values
- Updating the Host or Service with Changes

- Changing Recommended Values
- Configuring Special Tuner Agent Settings
- Upgrading a Tuning Agent
- Uninstalling a Tuning Agent

About Tuning Your System from the Console

LoadRunner Tuning Module's tuning features allow you to remotely tune hosts and services, from the Console machine. After ensuring that you have the necessary permissions and access rights on the host, you install a tuning agent—a small application—on the host. The tuning agent allows the Console to view the host's services, and to configure their settings.

Understanding the Configuration Tab

LoadRunner Tuning Module's Configuration tab contains the following:

- **Server Configurations**: A tree structure that displays the hosts and services you can tune. Hosts that are defined in the Topology window also appear in the Server Configurations tree, except for hosts that are used only as load generators. Note that you can add more hosts via the Configuration Tab (even if they are not defined in the Topology window).
- **Administration tab:** The Administration tab displays the properties of the hosts and services you select in the Server Configuration tree. It allows you to change those properties that are configurable, and to compare actual property values to recommended values and properties of other hosts and services. For information about the Administration tab, see "Viewing the Host Information," on page 671.
- Additional buttons that help you tune your system.
- Links to help topics that explain and guide you through the performance tuning process.

Supported Operating Systems

You can install tuning agents on computers running the following operating systems:

Windows: Windows NT, Windows 2000, Windows XP

UNIX: Solaris, HP, AIX, Linux

Applications That LoadRunner Tuning Module Can Tune

LoadRunner Tuning Module allows you to tune the following applications:

- Apache Web Server 1.x/2.x
- BEA Weblogic 6.x/7.x
- IBM HTTP Server
- IBM Websphere Advanced 4.x
- IBM WebSphere Application Server 5.x
- IBM Websphere Single Server 4.x
- iPlanet Enterprise Server 6 and higher
- Microsoft IIS 4/5
- Microsoft Active Server Pages 2/3
- Microsoft ASP.NET (Framework 1.0 & 1.1)
- Operating System (Windows: NT, 2000 and XP; UNIX: Solaris, HP, AIX and Linux)
- Oracle Database
- Oracle 9iAS
- PeopleSoft 8.x
- SAP Enterprise Portals 5
- Siebel 7.x
- SQL Server 7.5/2000

Tuning Flow

You tune each host in your system by following this procedure:

1. Configuring Host Connection Parameters. This involves specifying the settings that allow you to connect to the host machine and view its configuration information.
2. Connecting to the Host Computer. This also installs and starts a tuning agent. The tuning agent is an application that runs on the host machine, and communicates with the Console. You can install the tuning agent remotely from the Console machine or locally on the host.
3. Viewing the Host Information. After you connect to the host machine, you can view the information that is returned by the tuning agent.
4. Viewing and Changing Tuning Parameter Values. You change the values of the relevant host settings to enhance the host machine's performance.
5. Updating the Host or Service with Changes. This applies the new settings to the host machine.

Host Requirements

Before you install and run a tuning agent on the host machine, note the following requirements:

Java

The tuning agent requires a Java-enabled environment. Ensure that JRE/JDK 1.3 (or later) is installed on the host before you attempt to start the tuning agent. Either JRE or JDK can be used.

WMI Support

If the Console machine runs Windows NT, it must have Windows Management Instrumentation (WMI) support installed.
In addition, to enable remote installation of a tuning agent on a Windows NT host machine, you need to install WMI support on the host.

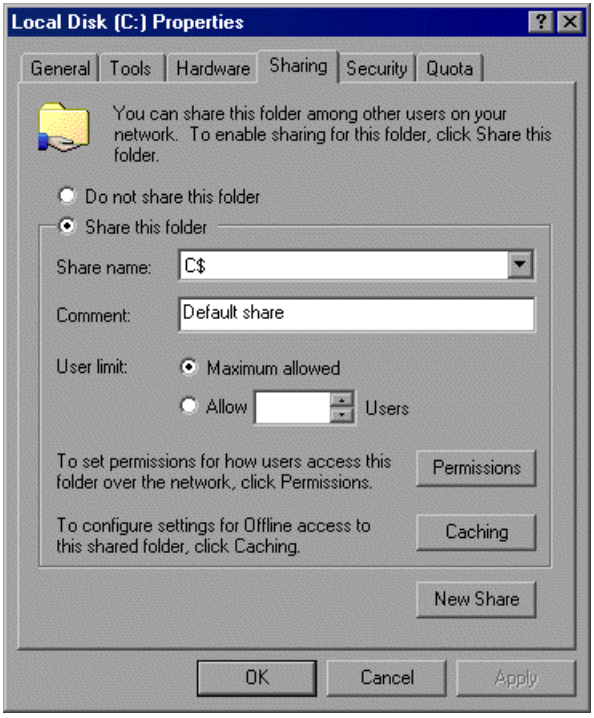
To install WMI, download it from
http://msdn.microsoft.com/library/default.asp?url=/downloads/list/wmi.asp

Drive Sharing

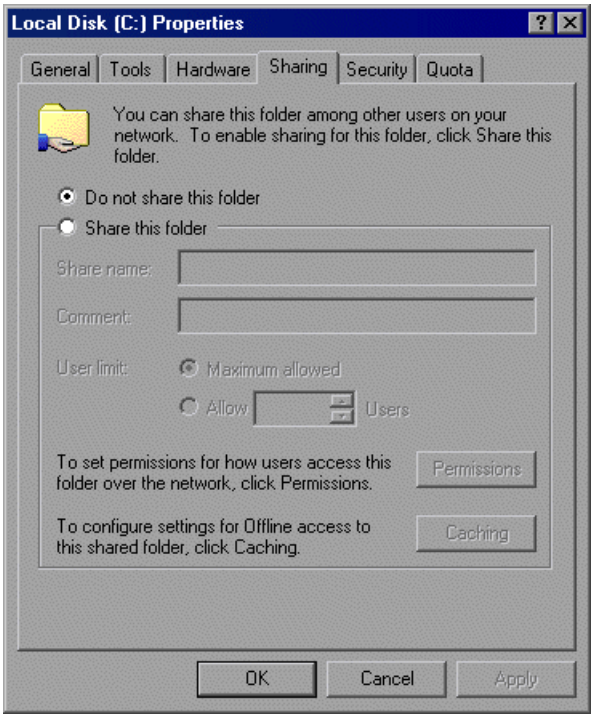
If the host that you are tuning is a Windows machine, ensure that its C: drive is shared with other users by default for administrative purposes.

To share the C: drive by default:

1 In Windows Explorer, right-click the C: drive and choose Sharing. The following dialog box appears. If the C: disk is already shared, the Share Name and Comment fields appear as follows:



2 If the C: disk is not shared by default, the **Share this folder** section is disabled and the Share Name and Comment fields are empty.



If this is the case, click **Share this folder** and enter C$ in the **Share name** list.

3 Click **OK**.

Tuning Through a Firewall

If there is a firewall between the Console machine and the machine that you want to tune, you need to open various firewall ports to enable you to remotely start the tuning agent and subsequently use it to tune the remote machine. The ports that you need to open depend on whether the remote machine is running Windows or UNIX.

Ports Needed for Starting a Tuning Agent

For Windows Machines:

- Port 135 (Client/Server Communication)
- Port 445 (CIFS—Common Internet File System)
- The port used by the Microsoft WMI service. This port is chosen dynamically by the service, and is one of the ports 1024 through 5120. Open the firewall's log file, determine the port through which the Console is attempting to connect, and open that port. Note that the port number can change when you restart the server.
- Port 4863 (SSL OTP) or 4862 (non-SSL OTP)

For UNIX Machines:

- The ports used by the remote shell (RSH). Open shell port (514) from the Console to the agent and auth port (113) from the agent to the Console. One of the ports 512 through 1023 is chosen dynamically by the RSH server (agent side).
- Port 4863 (SSL OTP) or 4862 (non-SSL OTP)

Ports Needed Once a Tuning Agent Has Been Started

Regardless of the remote machine's operating system, once you have started the tuning agent, only port 4863 or port 4862 needs to be open.

Configuring Host Connection Parameters

Before you install the tuning agent on the host machine that you want to tune, you specify the host and configure its connection parameters.

You need to perform the following actions:

1. Choosing the Host Machine
2. Specifying Tuner Agent Settings
3. Specifying Operating System Settings

Note: In addition, you can optionally specify SNMP settings.

---

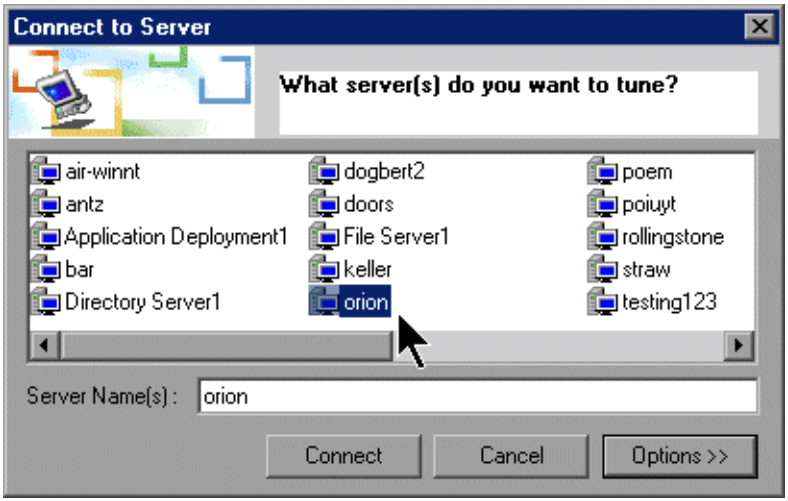
4. Connecting to the Host Computer

Choosing the Host Machine

To choose the host that you want to tune (if the host does not appear in the Servers Configuration tree):

➤ Click the **Connect to Host** button, or right-click the **Server Configurations** link on the left side of the window and click **Add New**.

The Connect to Server dialog box is displayed.

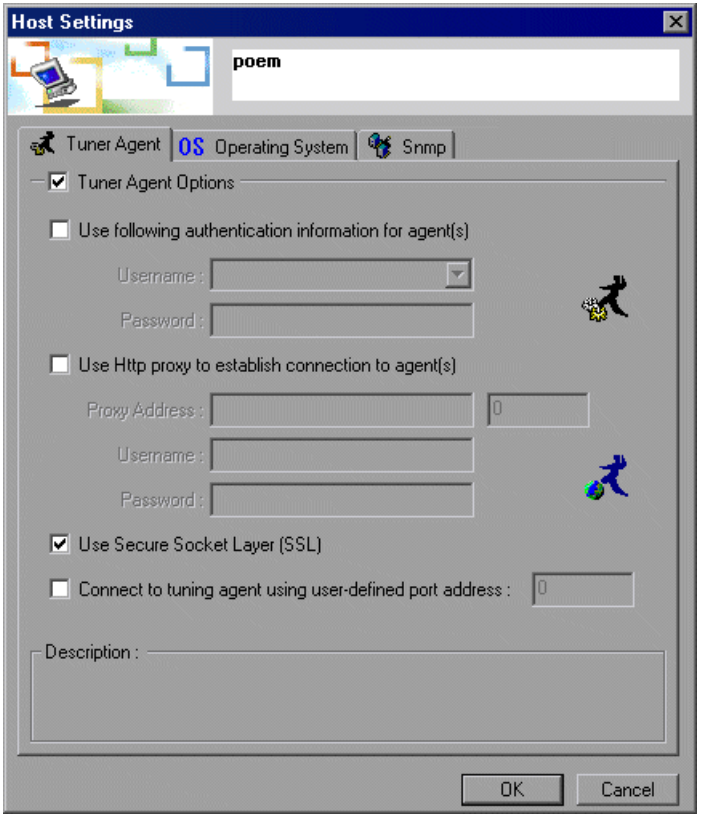


Specifying Tuner Agent Settings

Before you can connect to the host machine, you need to specify the tuner agent's settings. This allows you to start and stop the tuning agent, and to use it to tune the host machine.

To specify the tuner agent settings:

1 In the Connect to Server dialog box, click **Options**. Alternatively, for a host that you have already added to the Server Configurations tree (for example, if you added the host machine to your topology), click **View Host Properties**, or right-click the host's icon and choose Properties. The Host Settings dialog box is displayed.



2 In the **Tuner Agent** tab, enter the tuner username and password. This determines the actions that the tuner allows you to perform on the host. LoadRunner Tuning Module encrypts the password. You can select a username and password defined by the system administrator, or choose one of the following predefined username/password pairs:

Username	Password	Authorization	Comments
guest	(no password)	viewing access only	default
mercury	expert	viewing and update access	
admin	changeit	full administration access	The administrator of the tuner can define and grant tuning privileges to a user.

If you do not check the **Use following authentication information...** box, LoadRunner Tuning Module uses the *guest* username, allowing you only to query the host.

Note: For details on how to change passwords, see "Changing Tuning Agent Passwords," on page 700.

---

3 If the tuning agent on the host is accessible only through a proxy over a firewall, define the proxy settings in the **Use Http proxy...** dialog box.

4 If the agent is running (or will run) on a non-standard port (OTP-SSL 4863 or OTP 4862), you can specify a user-defined port. Check the **Connect to tuning agent** box and enter the port address in the relevant field.

5 Check the **Use Secure Socket Layer (SSL)** box to use SSL for all connections (recommended).

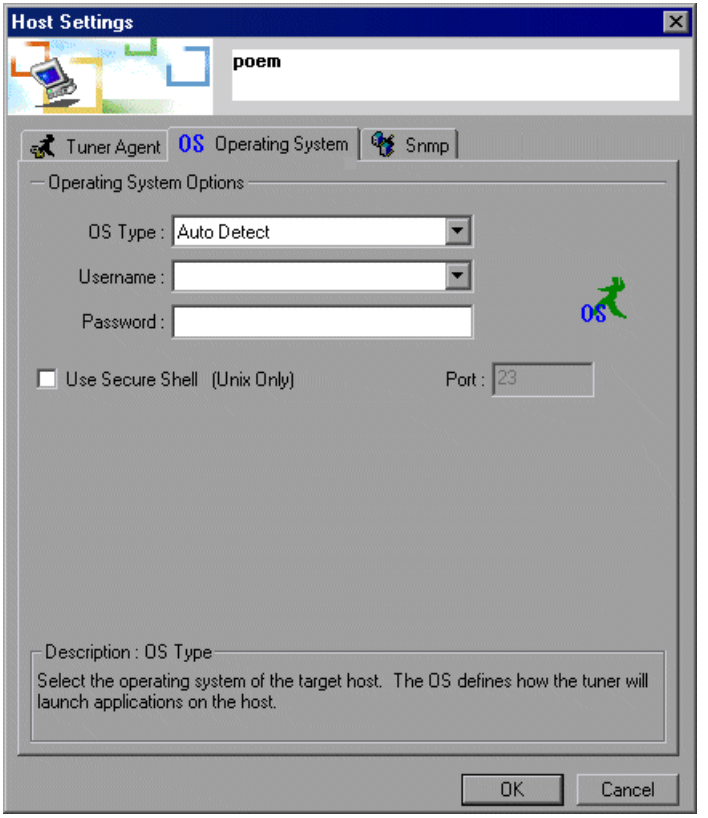
Specifying Operating System Settings

You need to specify the host machine's operating system and username/password information, so that the host machine will allow you to access it via the tuner agent.

To specify the operating system settings:

1 In the Connect to Server dialog box, click **Options**. Alternatively, for a host that you have already added to the Server Configurations tree (for example, if you added the host machine to your topology), click **View Host Properties**, or right-click the host's icon and choose Properties. The Host Settings dialog box is displayed.

2 Click the **Operating System** tab.



3 Choose the host's operating system from the list in the OS Type box.

4 Enter the host's username (including the domain) in the Username field. For example, if your computer's name is straw and your username is joe, enter the string straw\joe.

5 Enter the host password in the Password field.

6 To use secure shell (for UNIX only), check the **Use Secure Shell** box and enter the port address in the Port field.

Note: After you have specified the tuning agent and operating system settings (and, optionally, SNMP settings), you need to connect to the host computer, as described in the following section.

Connecting to the Host Computer

After configuring the tuning agent settings, you need to connect to the host computer.

To connect to a newly-added host computer:

1 In the Host Settings dialog box, click **OK** to save your settings and close the dialog box.

2 In the Connect to Server dialog box, ensure that the host appears in the Server Name field.

3 Click **Connect**.

This installs the tuning agent on the host machine and starts it.

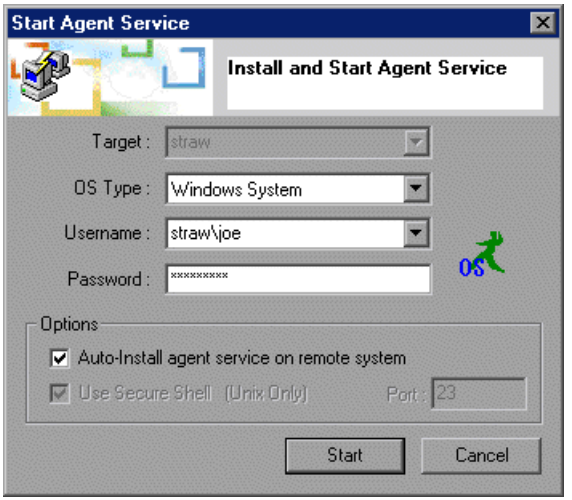
If you added a host machine to the Server Configurations tree but did not start the tuning agent (for example, if you added the host machine to your topology), use one of the following procedures to install and start the tuning agent.

Remote Installation of a Tuning Agent

To remotely install a tuning agent from the Console machine:

1 Click the host's icon in the Server Configurations tree, and then click the **Start Tuning Agent** button on the toolbar.

The Start Agent Service dialog box is displayed:



2 Specify the host's OS Type. To let LoadRunner Tuning Module automatically detect the operating system, choose Auto Detect.

3 Check the **Auto-Install...** box.

4 If you are installing the tuning agent on a UNIX machine, check the **Use Secure Shell** box and specify the port address, if applicable.

5 Click **Start**. If you've configured your settings correctly, LoadRunner Tuning Module installs the tuning agent on the host and starts the javaw.exe process on it.

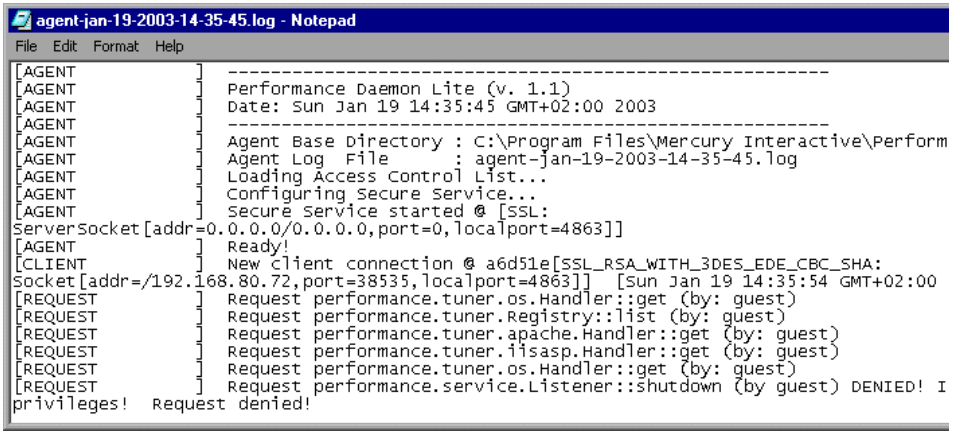
Note: On a Windows host, the tuning agent is installed under C:\Program Files\Mercury Interactive\Performance Expert. On a UNIX host, it is installed under /tmp/perfexpert.

On the Console machine, the host's icon in the Server Configurations tree changes to blue, indicating that the connection to the host is alive.

The tuning agent log file is located under \<Tuning agent installation directory>\agent. It keeps track of the following activity occurring during the tuning session:

- Client connections
- Client requests (get, set, start, stop)
- Parameters that are modified, and their new values
- Errors

Following is an example of the log file:



If the tuning agent does not start, or doesn't show the expected information, you may need to configure its settings. See "Configuring Tuning Agents," on page 699.

Local Installation of a Tuning Agent

To install a tuning agent locally from the LoadRunner Tuning Module CD on a Windows host:

- Choose the **Tuner Agent** action in the installation procedure.

Note: After you install the tuning agent locally from the CD, you can start the agent by choosing **Start** > **Programs** > **Performance Expert** > **Tuning Agent** on the host machine.

To install a tuning agent locally on a Windows host:

1. Extract the perfagent.tar file (located in the *\LoadRunner Tuning Module\console\bin* directory) to a directory (for example, to *C:\Program Files\Mercury Interactive\Performance Expert*).
2. Set the PE_HOME environment variable (in this example, PE_HOME = C:\Program Files\Mercury Interactive\Performance Expert).
3. Launch the tuning agent batch file. In this example, you would run the following file:

   C:\Program Files\Mercury Interactive\Performance Expert\agent\bin\pe_agent.bat

   You can skip step 2 by passing the path to the installation directory to pe_agent.bat in the command line, as in the following example:

   % pe_agent.bat 0 true C:\Program Files\Mercury Interactive\Performance Expert

To install a tuning agent locally on a UNIX host:

1. Copy the perfagent.tar file (from the *\LoadRunner Tuning Module\console\bin* directory on the Windows machine where the Console is installed) to a directory on the UNIX machine (for example, to */usr/local/perfexpert*), and extract it to that directory.
2. Set PE_HOME environment variable (in this example, setenv PE_HOME /usr/local/perfexpert -- for CSH, ...)
3. Launch the tuning agent batch file. In this example, you would run the following file:

   /usr/local/perfexpert/agent/bin/pe_agent

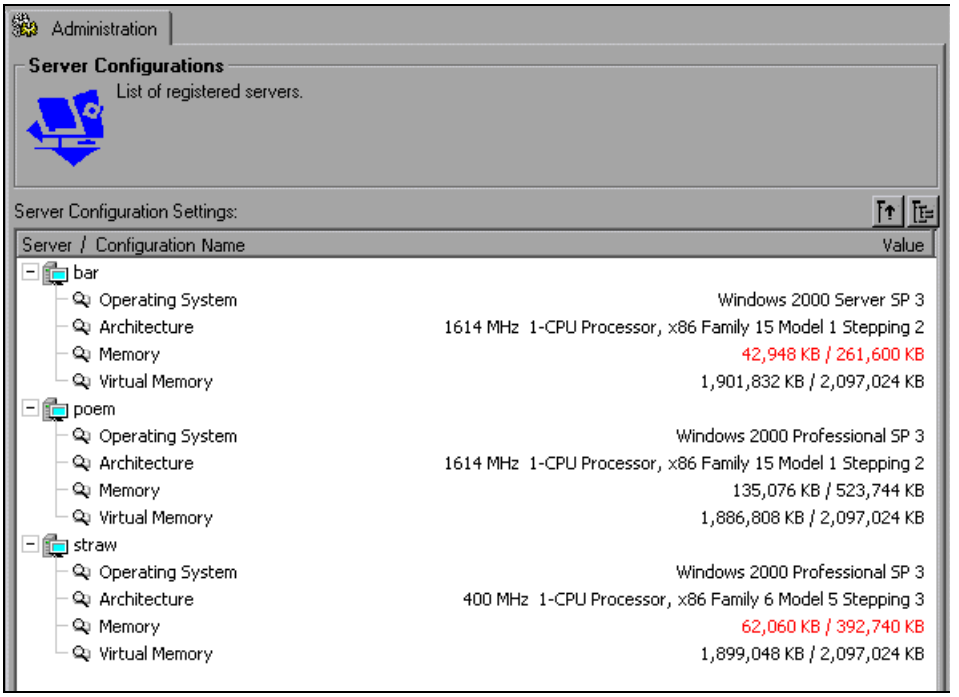
   You can skip step 2 by passing the path to the installation directory to pe_agent in the command line, as in the following example:

   % pe_agent 0 true /usr/local/perfexpert

Viewing the Host Information

The Administration tab shows you the host property information, and allows you to change the values of those properties that are configurable.

After you've connected to the host and the tuning agent is running, click the Server Configurations element. The Administration tab shows summary information for each of the hosts, as in the following example:



The following information is displayed about each host:

- Hostname
- Operating System
- Architecture—CPU type, including number of processors
- Memory—available RAM over total RAM. If the amount of available RAM is less than 20% of the total RAM and is less than 100MB in size, it is displayed in red.

➤ Virtual Memory—current available virtual memory over total virtual memory (pagefile). If the amount of virtual memory is less than 20% of the total virtual (swap) memory, it is displayed in red.

When you click the element of a host or service in the tree, the Administration tab shows information about the host or service, including parameter values.

To expand all the nodes in the Administration tab, click **Expand All Nodes**.

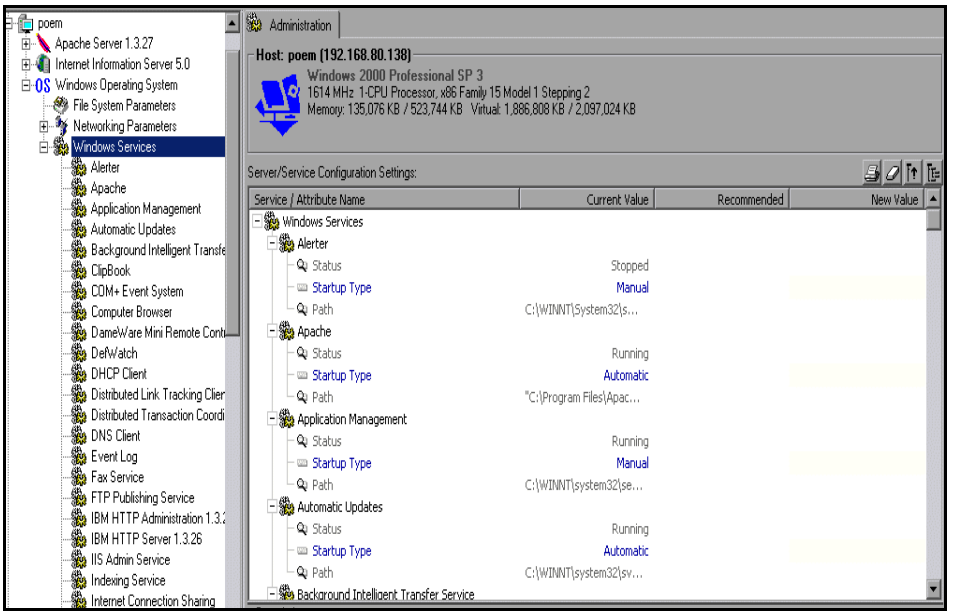
To collapse all the nodes in the Administration tab, click **Collapse All Nodes**.

Viewing Windows Services

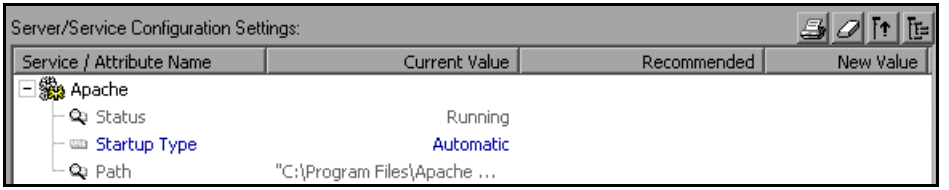
When the host machine is running Windows, a Windows Services element appears as a sub-element of the Windows Operating System element. You should use this element to stop and start the Services settings on hosts with Windows, instead of using the File System Parameters element.

Note: For details of how to start and stop a Windows service, see "Configuration Tab Functions," on page 715.

Expanding the Windows Services element displays the host machine's services.



When you click a service in the Windows Services tree, LoadRunner Tuning Module displays the service's status in the Configuration tab.



The following service properties are displayed:

- Status—**Running**, **Stopped** or **Paused**
- Startup Type—**Automatic** (starts automatically when Windows is started), **Manual** (must be started manually by the user) or **Disabled**.

Note: You can change the startup type via the Configuration tab (see "Viewing and Changing Tuning Parameter Values," on page 677).

➤ Path—The path to the service

Using Expert Mode

Some tuning parameters are displayed only if you are in Expert mode.

To enable Expert mode:

➤ Click the icon of the host or service for which you want to enable it, and then choose **Expert View** from the box in the Configuration tab's toolbar. Alternatively, you can right-click the host or service and choose **View Mode** > **Expert View**. Expert mode is enabled for the selected node and its sub-nodes, and the extra parameters are displayed.

To disable Expert mode:

➤ Choose **Normal View** from the box in the toolbar, or right-click the host or service and choose **View Mode** > **Normal View**. This disables Expert mode for the selected host or service.

Viewing Differences Only

The Differences Only view allows you to view only those parameters whose values differ from the relevant recommended values.

To enable the Differences Only view:

➤ Choose **Differences Only** from the box in the Configuration tab's toolbar. Alternatively, you can right-click the host or service and choose **View Mode** > **Difference View**.

Comparing Hosts and Services

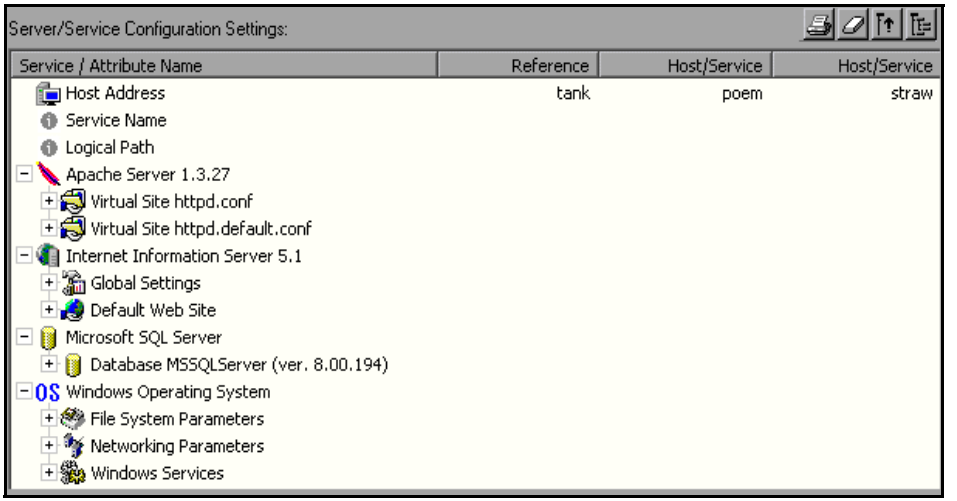
LoadRunner Tuning Module allows you to compare the values of properties of different hosts and services, and shows you the properties that are different.

To compare hosts or services:

1 In the Server Configuration tree, click the hosts or services that you want to compare (use the <Ctrl> key to select multiple hosts or services). Note that the last host or service that you select will be used as the reference for the comparison. This means that all the services on the reference will be compared to those on the other hosts or services.

2 If the Configuration tab is not displayed, click it.

3 Click **Compare Configuration Settings**. The Configuration tab shows a column of properties for each of the servers or services being compared.



4 Expand the service whose properties you want to view, or click **Expand All Nodes**to expand all the services. LoadRunner Tuning Module displays the properties of the reference and the other hosts and services with which you're comparing it.

Service / Attribute Name	Reference	Host/Service	Host/Service
Host Address	tank	poem	straw
Service Name			
Logical Path			
Apache Server 1.3.27			
Virtual Site httpd.conf			
Port	81	80	80
Log Level	warn	n/a	warn
Keep-Alive Connections	ON	ON	ON
Keep-Alive Timeout	60	15	15
Max Keep-Alive Requests	100	100	100
Max Requests Per Child	0	0	0
Send/Recv Timeout	120	300	300
Threads Per Child	200	50	50
Virtual Site httpd.default.conf			
Internet Information Server 5.1			
Microsoft SQL Server			
Windows Operating System			

Note that properties whose values are different on the different servers are highlighted on the screen.

5 To locate the next property whose values differ, click **Show Next**.

6 To locate the previous property whose values differ, click **Show Previous**.

7 To clear all new values of a selected host or service, click **Clear all new values in the configuration**.

8 To print the configuration settings of a selected host or service, click the **Generate HTML report** button. The HTML report is displayed by your browser. Print the report from the browser.

Viewing and Changing Tuning Parameter Values

You use the Configuration tab to view and change values of tuning and configuration parameters for the selected host or service. For each parameter, LoadRunner Tuning Module displays the following:

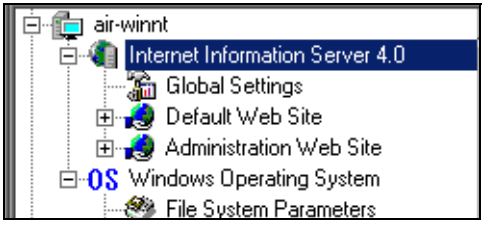
- **Current value**
- **Recommended value**—the value recommended by LoadRunner Tuning Module
- **New value**—the value that you entered (if you changed the old value)

The different parameters are color-coded as follows:

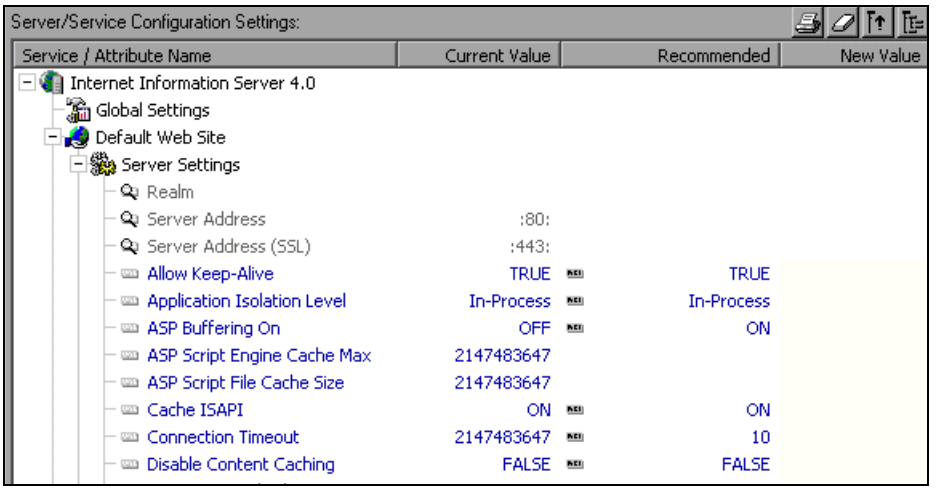
Color	Meaning
blue	important
black	for advanced users
red	critical tuning parameter
gray	read-only

To change the value of a host parameter:

1 Select the host or service that you want to tune, by clicking the relevant icon in the Server Configuration tree.

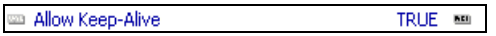


In the Administration tab, LoadRunner Tuning Module displays the parameters relevant to the selected host or service.



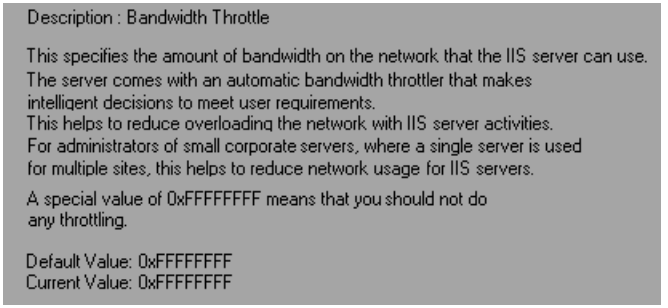
The values in the Recommended column are based on information in LoadRunner Tuning Module's knowledge base.

If the Tuning Module's knowledge base contains a recommended value for a parameter, the Recommended Value icon appears adjacent to the current value, as in the following example:

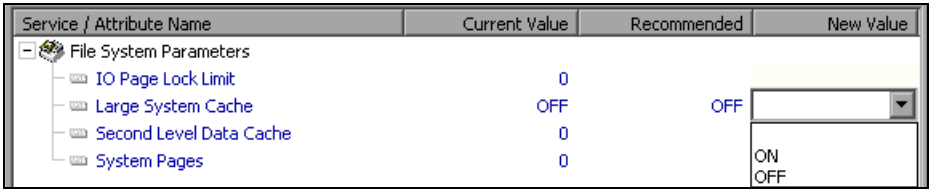


If the Tuning Module's knowledge base contains a list of recommended values for a parameter, the Recommended Value List icon appears adjacent to the current value, as in the following example:

When you click a property, the lower section of the Administration tab displays a description of the property and its values, and may include tuning recommendations.



2 Click the parameter that you want to configure, and click the parameter's New Value column. If the parameter is configurable, a text box opens, or a list box appears.



3 Enter the new value. In the case of a list box, choose it from the list.

4 Update the host or service (see "Updating the Host or Service with Changes," on page 679). When you refresh the view, the new value appears in the Current Value column.

To clear all new values of a selected host or service, click the **Clear all new values in the configuration** button.

Updating the Host or Service with Changes

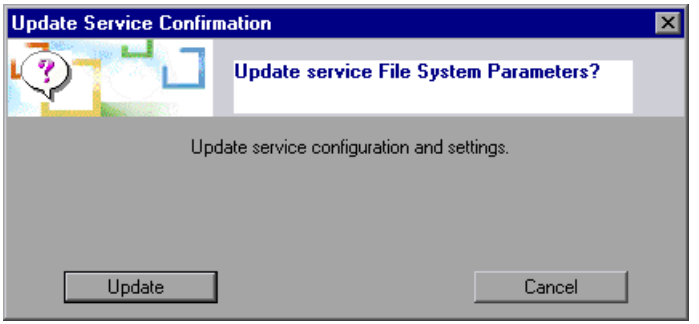
Changes that you make to parameter values do not take effect until you update the relevant host or service.

When you update a host, all of the services in the host's tree are updated (if any of the parameters have been assigned new values).

When you update a service or service category, only the selected service or category is updated.

To update a host or service:

1 Click **Update changes to Service/Host**, or right-click the host or service icon in the Server Configurations tree and click **Update**. The Update Service Confirmation dialog box is displayed.



2 Click **Update** to update the selected host or service. Note that some services need to be restarted for the changes to take effect.

LoadRunner Tuning Module displays a message informing you that the changes have taken effect.

3 To view the changed values, right-click the host icon and click **Refresh**.

Changing Recommended Values

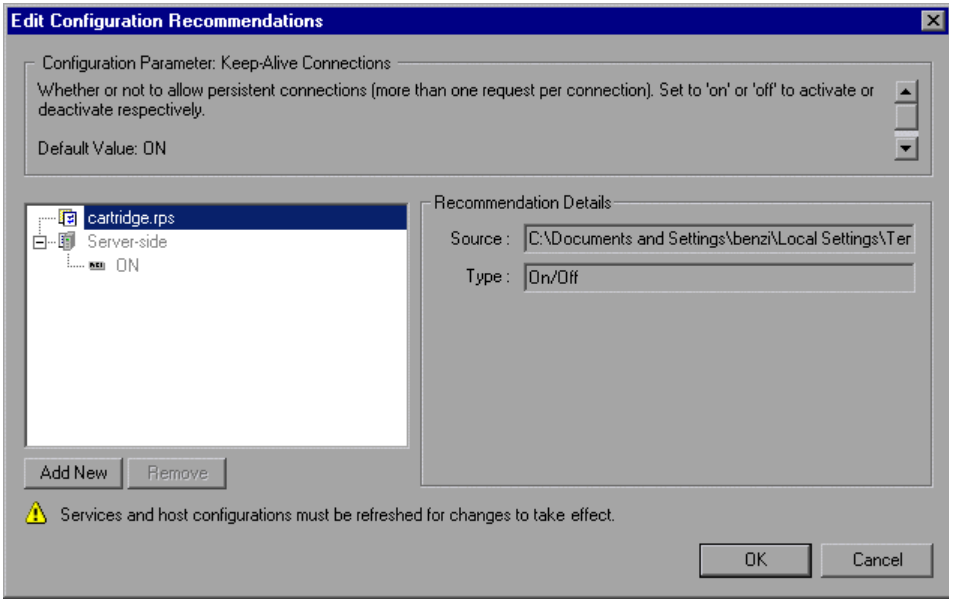
LoadRunner Tuning Module allows you to add a value to a parameter's list of recommended values, or to remove a value from the list. You can also edit an existing value.

Note: Before you add or remove a recommended value, you need to enable the cartridge functions (**Cartridge** > **Enable Cartridge Functions**).

To add or remove a recommended value:

➤ Click the Recommended column of the parameter whose list of recommended values you want to change, and choose **Edit** from the drop-down list.

The Edit Configuration Recommendation window is displayed.



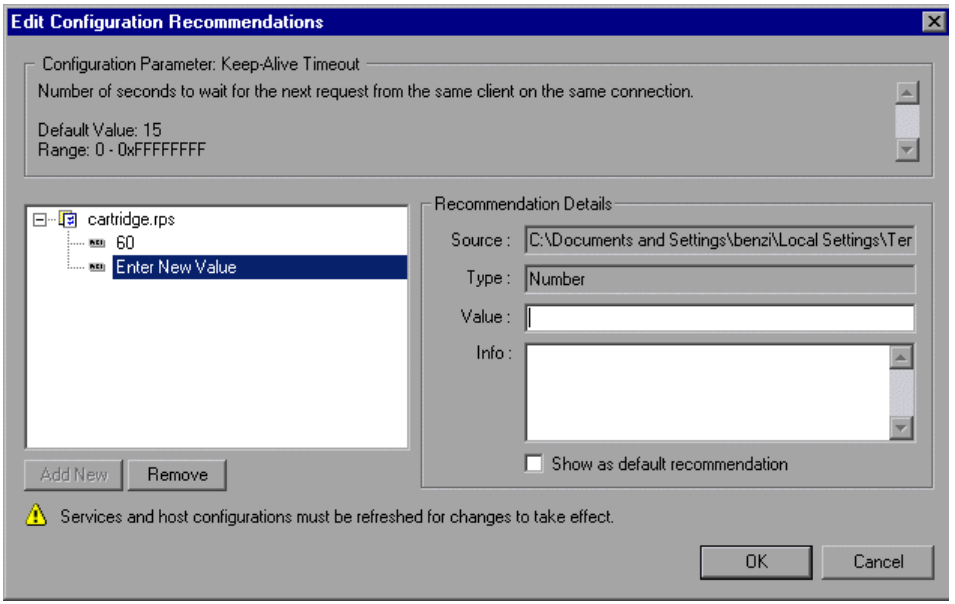
If there are recommended values for the selected parameter, they are displayed in the left pane.

Note that you cannot edit server-side recommendations (recommendations generated by the tuning agent on the host machine) via the Edit Configuration Recommendations window. You can only edit client-side recommendations.

If recommended values have been defined on the server side, they will appear in this pane as read-only values.

To add a new recommended value:

1 Click **Add New.** The Value and Info fields appear in the Edit Configuration Recommendations window.



2 Enter the new value in the Value column, or choose it from the drop-down list.

If the value is numeric, you can specify it using one of the following conditional expressions:

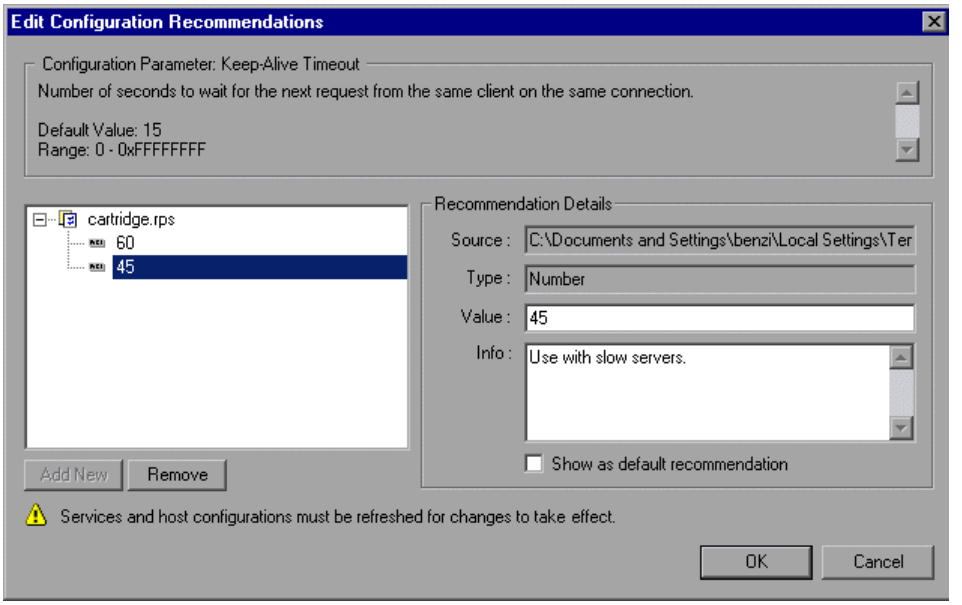
Condition	Syntax
Value is greater than X	>X , #>X , X< , X<#
Value is greater than or equal to X	>=X , #>=X , X<= , X<=#
Value is less than X	<X , #<X , X> , X>#
Value is less than or equal to X	<=X , #<=X , X>= , X>=#
Value is greater than X and less than Y	X<#<Y , Y<#<X

Condition	Syntax
Value is greater than or equal to X and less than Y	X<=#<Y , Y<#<=X
Value is greater than X and less than or equal to Y	X<#<=Y , Y<=#<X
Value is greater than or equal to X and less than or equal to Y	X<=#<=Y , Y<=#<=X

3. In the Info field, enter the text explaining when you should use this value. This text will be displayed in the Description field of the Administration tab, when the value is selected.
4. To cause the new value to be displayed as the recommended value (that is, as the first value in the list), check **Show as default recommendation**.
5. Click **OK** to close the Edit Configuration Recommendations window.
6. Right-click the server or service icon in the Server Configuration tree, and choose **Refresh**. This adds the value to the list of recommended values for the parameter.

To remove a recommended value:

1 In the Edit Configuration Recommendation window, select the value that you want to remove, as in the following example:



2 Click **Remove**. The value is deleted from the displayed list in the Edit Configuration Recommendations window.

3 Click **OK** to close the Edit Configuration Recommendations window.

4 Right-click the server icon in the Server Configuration tree, and choose **Refresh**. This removes the value from the list of recommended values for the parameter.

Configuring Special Tuner Agent Settings

You can configure the following special types of tuner agent settings:

➤ SNMP Settings

Enabling SNMP For Tuning

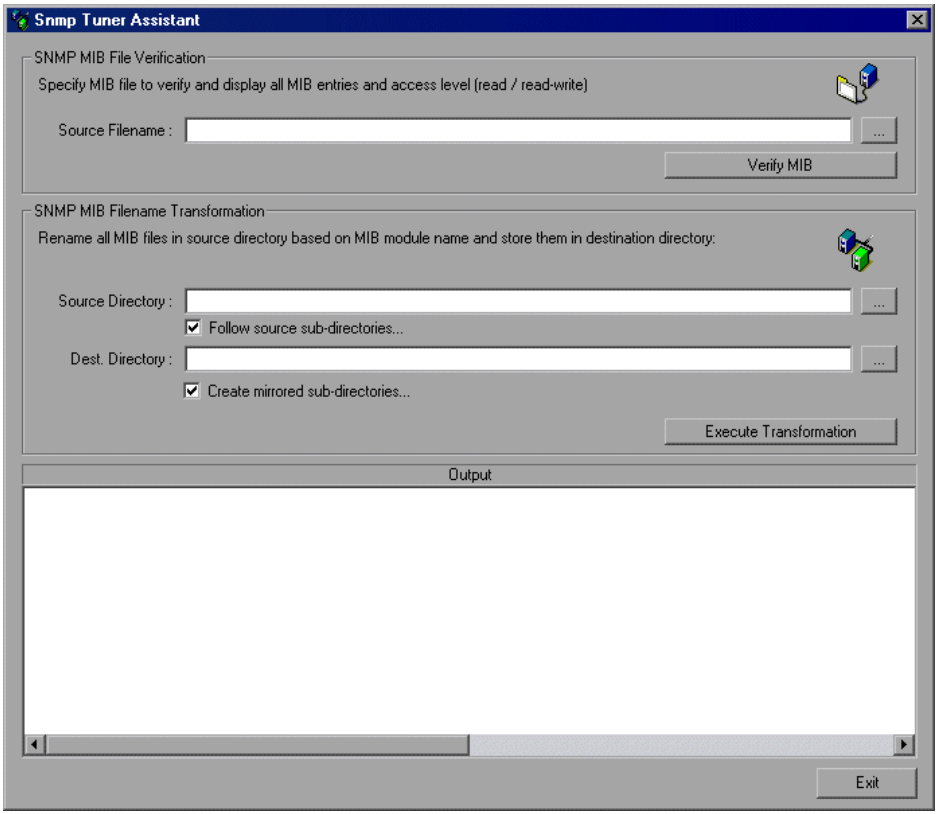
LoadRunner Tuning Module allows you to use SNMP to tune host machines. Note that the SNMP service must be running on the host machine.

To enable SNMP for tuning a host machine:

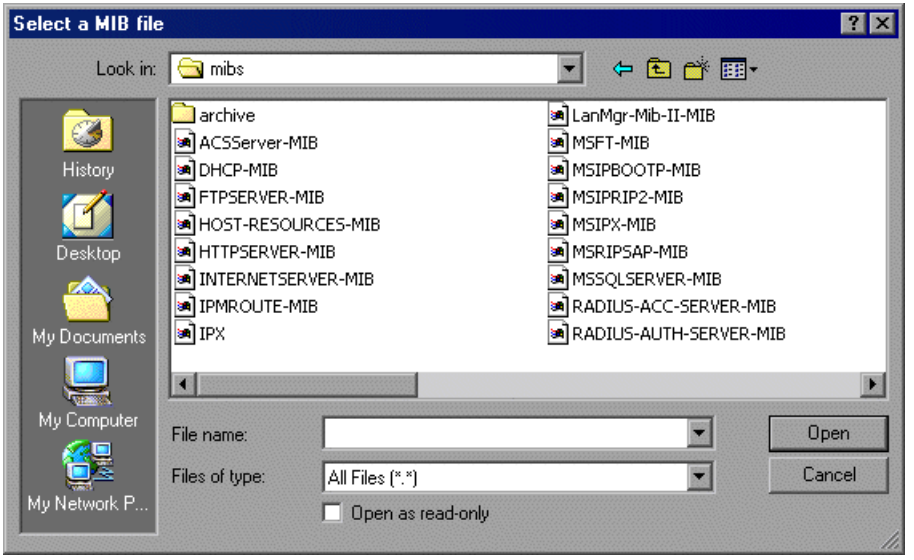
1 The LoadRunner Tuning Module installation directory contains zip files for connecting to different applications via SNMP. These files are located in the \console\dat\snmp\mibs\archive subdirectory.

Locate the zip file that applies to the applications to which you want to connect, and extract it to the \console\dat\snmp\mibs directory. (For example, if you want to connect to Microsoft applications, extract the file Microsoft.zip.) Ensure that you do not create a directory when extracting the file.

2 In the console\bin directory, double-click the file snmpasst.exe. The SNMP Tuner Assistant dialog box is displayed.



3 Using the Browse button at the right of the Source Filename field, browse to the console\dat\snmp\mibs directory. The list of .MIB files in the directory is displayed.



4 Select the file for the protocol that you want to use, and click **Open**. The filename now appears in the Source Filename field.

5 Click **Verify** to check that the MIB file has not been corrupted.

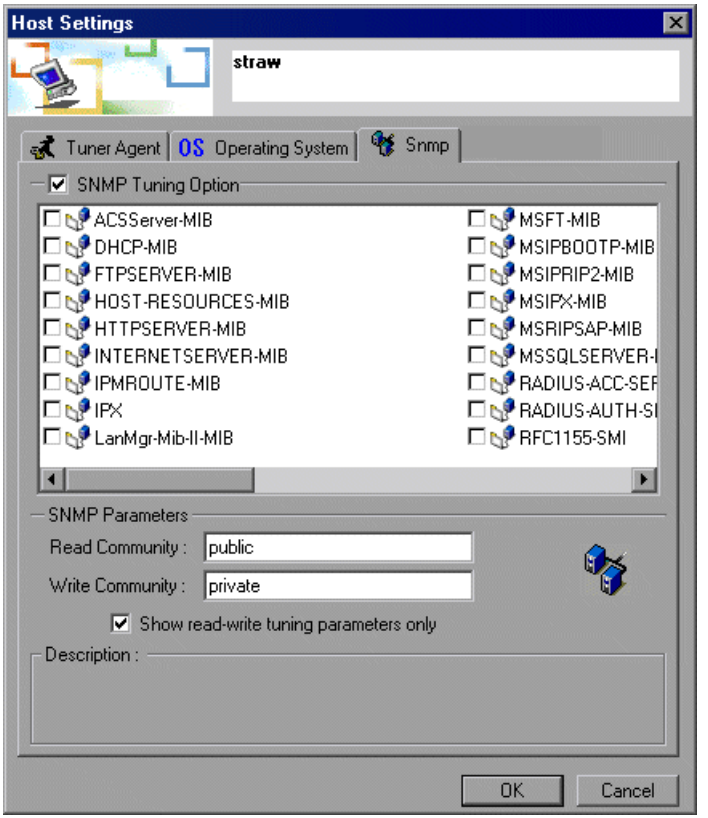
6 If error messages are displayed in the Output window, edit the .MIB file and correct the errors. Then click **Verify** to check the file again.

7 When the file has been successfully verified, click **Exit** to close the SNMP Tuner Assistant.

To specify SNMP settings:

1 In the Connect to Server dialog box, click **Options**. Alternatively, for a host that you have already added to the Server Configurations tree, click **View Host Properties**, or right-click the host's icon and choose Properties. The Host Settings dialog box is displayed.

2 Click the **SNMP** tab, and check **SNMP Tuning Option**.



3 Select the MIB files that you need for tuning the host machine via SNMP. For example, if you want to tune the host machine's MS-SQL Server, select MSSQLSERVER-MIB.

4 In the Read Community field, specify the community to use for retrieving values from the server. Default: public.

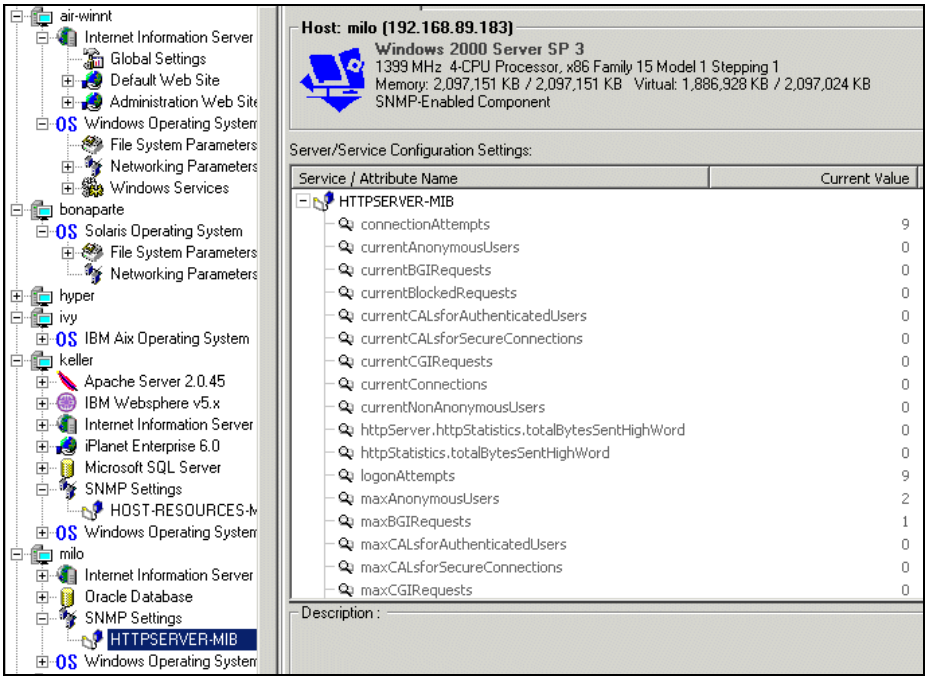
5 In the Write Community field, specify the community to use for writing new values to the server. Default: private.

Note: If a service is not available through the default public and private communities, you need to specify the appropriate community. Please check the server's SNMP configuration, or check with your administrator, for the correct read and write communities.

---

6 To cause the Configuration tab to display only changeable tuning parameters, check **Show read-write tuning parameters only**.

After you apply your changes and exit the Host Settings dialog box, the SNMP Settings service is added to your host in the Server Configurations tree, with the .MIB file that you chose.



Upgrading a Tuning Agent

To upgrade a tuning agent with a new version:

1. Stop the tuning agent (see "Stop the Tuning Agent," on page 718).
2. Remove the tuning agent from the host machine (see "Uninstalling a Tuning Agent" on page 690 below).
3. Install the new tuning agent remotely from the Console machine, or install it locally on the host machine. For details of how to install a tuning agent, see "Connecting to the Host Computer," on page 667.

Uninstalling a Tuning Agent

The way you remove a tuning agent from a server depends on the way it was originally installed.

If you installed the tuning agent from the CD:

➤ Select **Start** > **Programs** > **Performance Expert** > **Uninstall Performance Expert Installation** and follow the onscreen instructions.

If you installed the tuning agent remotely from the Console machine, or by copying files from the Console machine to the remote machine:

1. Stop the tuning agent (see "Stop the Tuning Agent," on page 718).
2. On the host machine, delete the tuning agent files from the appropriate location:

➤ On a Windows machine, delete the directory C:\Mercury Interactive\Performance Expert\agent. If you don't find the files there, you can find their location by checking the PE_HOME environment variable. Then delete the PE_HOME environment variable.

➤ On a UNIX machine, delete the directory /tmp/perfexpert.

46

Exporting and Importing Configuration Settings

This chapter describes how to export and import configuration settings.

It includes the following topics:

- Exporting a Host or Service's Configuration Settings
- Importing Configuration Settings for a Host or Service
- Saving and Loading Profiles
- Creating a New Profile

About Exporting and Importing Configuration Settings

Once you've viewed (and possibly changed) configuration settings on your hosts and services, you can export the settings to save them for use in the future. You can also import previously saved settings to hosts and services.

Exporting settings allows you to track changes on the hosts and services, and to import the settings into other devices.

Importing previously saved settings for a host or service allows you to replicate desired settings across multiple hosts with similar configurations, avoiding the need to manually update the settings on each host or service individually.

You can export and import settings for a service, a host, or a group of hosts. When you export the settings of a group of hosts, you create a profile. You can subsequently import the profile.

Exporting a Host or Service's Configuration Settings

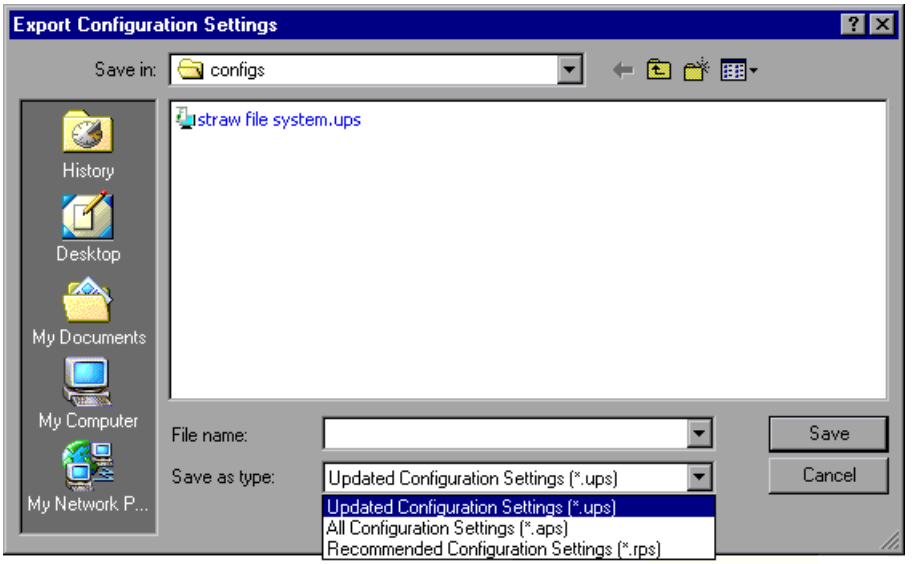
The Export function allows you to store the configuration settings of the entire host, or only a subset of these settings (for example, the settings for a specific service). You can export the following sets of values:

- All the settings, including those that have not been changed.
- Updated or recommended settings. This means exporting only the settings to which you've assigned new values but have not committed (that is, values that appear in the New Value column). When you subsequently import the settings, LoadRunner Tuning Module handles them as follows:
  - When you import a recommended settings file (a file with the *.rps* extension), LoadRunner Tuning Module inserts the values in the Recommended column.
  - When you import an updated settings file , (a file with the *.ups* extension), LoadRunner Tuning Module inserts the values in the New Value column.

See "Updating the Host or Service with Changes," on page 679 for details of how to commit changes.

To export configuration settings:

1 Click the host or service whose values you want to export and then click **Save/Export configuration settings**. Alternatively, you can right-click the host or service and click Export Settings. The Export Performance Settings dialog box is displayed.



2 From the Save as Type box, choose whether to save only the updated (and uncommitted) settings, the recommended settings, or all the settings for the selected host or service. If you save only the updated settings, LoadRunner Tuning Module saves them in a file with a *.ups* extension; the recommended settings are saved in a file with an *.rps* extension; if you choose to save all the settings, they are saved in a file with an *.aps* extension.

3 Enter a meaningful name for your settings file and click **Save**.

LoadRunner Tuning Module saves your settings in a file with the name you specified.

Importing Configuration Settings for a Host or Service

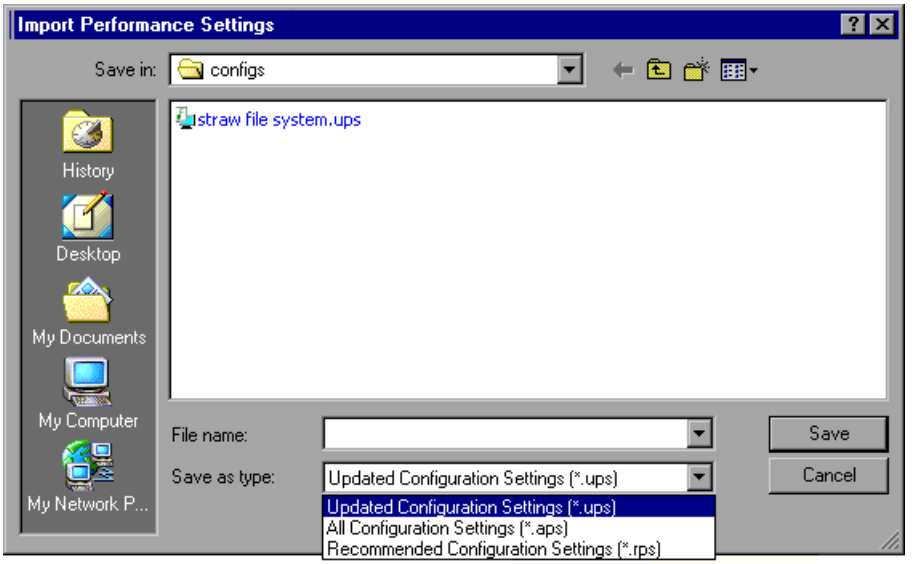
The Import function allows you to import previously stored settings and apply them to hosts and services. You can import and apply settings to:

- Hosts
- Services
- Sub-categories of services

Only the entries that are relevant to the host, service or sub-category into which you are importing will be imported from the settings file. For example, you may have a file that contains settings for both Internet Information Server (IIS) and Windows Operating System. If you click the host's Windows Operating System tree element and then import the settings from the file, LoadRunner Tuning Module loads only the network settings for Windows Operating System, not the IIS settings. Another example: Importing Apache settings into a host that includes IIS does not overwrite the host's IIS settings.

To import configuration settings:

1 Click the host or service whose values you want to import and then click **Load/Import configuration settings**. Alternatively, you can right-click the host or service and click Import Settings. The Import Performance Settings dialog box is displayed.



2 From the Files of type box, choose whether to view the files containing only updated performance settings (files with *.ups* extensions), recommended settings (files with *.rps* extensions), or files containing all the performance settings for the selected host or service (files with *.aps* extensions).

3 Select the settings file from which you want to import, and click **Open**.

LoadRunner Tuning Module imports the settings into the specified host, service or category.

4 Before the imported settings take effect, you need to commit them (see "Updating the Host or Service with Changes" on page 679). This prevents you from accidentally importing configuration settings, and allows you to inspect, validate, and possibly fine-tune the imported settings before updating the host or service.

Saving and Loading Profiles

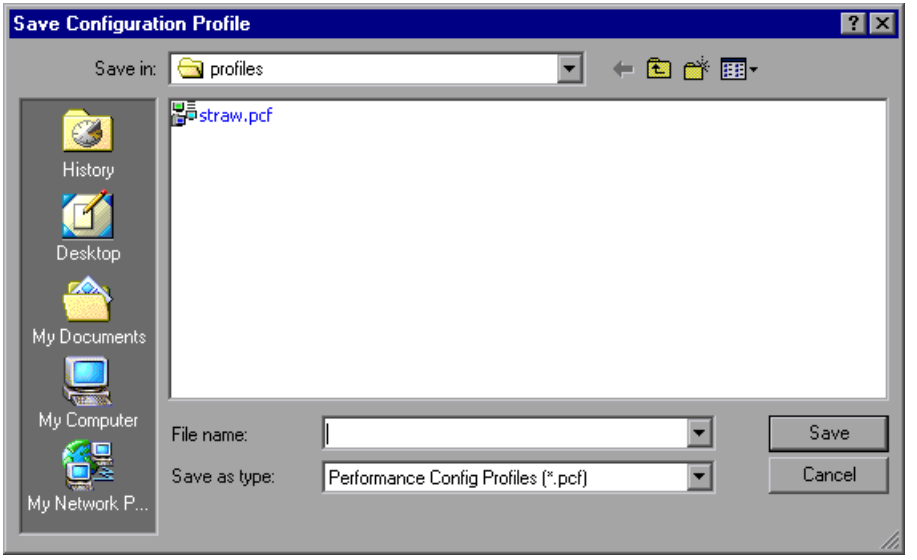
A profile contains a list of hosts.

When dealing with large clusters of hosts that may have different operational functions, creating profiles lets you save and load configuration settings for all the hosts in the group. For example, you may find it useful to group a cluster of Web servers into a profile, or group a set of servers relating to an e-commerce or intranet application.

When you save a profile, LoadRunner Tuning Module saves the list of hosts that appear in the Configuration window into a file for retrieval at a later time.

To save a profile:

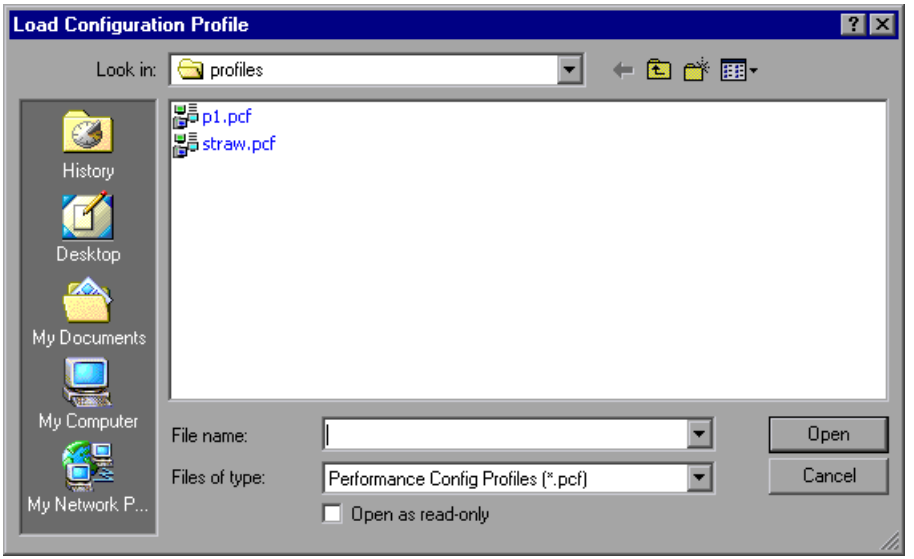
1 Click **Save Tuning Profile**, or right-click the Server Configuration tree element and click Save Profile. The Save Configuration Profile dialog box is displayed.



2 Enter a name for your profile and click **Save**. LoadRunner Tuning Module saves the profile with a *.pcf* extension.

To load a profile:

1 Click **Open Tuning Profile**, or right-click the Server Configuration tree element and click Load Profile. The Load Configuration Profile dialog box is displayed, showing the saved profiles as *.pcf* files.



2 Choose the profile you want to load, and click **Open**. LoadRunner Tuning Module loads the profile and connects to each of the profile's hosts to retrieve and load the profile's settings into the current host configuration. LoadRunner Tuning Module displays the new configuration in the Configuration tab.

Creating a New Profile

Note: When LoadRunner Tuning Module creates a new profile, it erases the current profile from memory. This means that all the server settings that you defined will be cleared from the window. If you need to save your settings, make sure you save the existing profile before creating a new one.

To create a new profile:

1 Click **New Tuning Profile**. LoadRunner Tuning Module removes the server icons and their accompanying information from the Configuration tab.

2 Add the required servers and services, and connect to them.

3 Save the new profile.

47

Configuring Tuning Agents

A tuning agent is an application that runs on the host that you want to tune, and allows you to tune the host remotely. This chapter describes the tuning agent's features and explains how to configure its settings.

The chapter includes the following topics:

- Changing Tuning Agent Passwords
- Changing the Tuning Agent's Port
- Using the Performance Tuner Registry
- Automatically Starting the Tuning Agent when Booting
- Starting and Stopping the Apache and IBM HTTP Servers

About Configuring Tuning Agents

The tuning agent interrogates the host and gathers performance-related information and tuning parameters, and passes the information to the Console. It allows the user to remotely configure and administer the target system.

The tuning agent is a passive service and does not consume any CPU resources when not processing requests. The agent requires only between 10 and 15MB of memory on the host.

The Console uses the tuning agent to change the target system's configurable parameters.

You install the tuning agent over the network from the LoadRunner Tuning Module Console workstation, or locally on the target system (see "Connecting to the Host Computer," on page 667).

The tuning agent does not require any registry updates. It uses the PE_HOME, PE_USE_SSL, and PE_USE_PORT optional environment variables. Alternatively, you can pass these as arguments to the pe_agent and pe_registry commands. For information on these commands, see "Configuring Tuning Agents," on page 699.

Changing Tuning Agent Passwords

The tuning agent is supplied with a number of predefined users and passwords (see the table in "Specifying Tuner Agent Settings," on page 664). It is recommended to change these passwords to prevent unauthorized access to the tuners.

The passwords are defined in the security.properties file, located in the Performance Expert\agent\config directory on the host computer. Note that the passwords must be specified separately for each host.

Following is an example of the relevant section in the security.properties file:

```
# -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
# List of users
# -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
# List of active users that are allowed access to the performance expert ager
# service.  By default, admin, mercury, and guest users are supplied.  The
# guest user is used if client does not supply any credentials.  In order to
# enforce strict user authentication and control, you should remove the guest
# user account from the user list (below).
#
# Format :  user.<NAME>=<PASSWORD>
#
# Example:  user.admin=changeit
#
# It is strongly recommended that the passwords for admin, mercury and guest
# be changed.  The passwords for users will automatically be encrypted when
# tuning agent process (pe_agent) is started.
#
user.admin={DES3S}5197355F27E7E24C28F2D2B9B63C8B2E
user.guest={DES3S}FFD6868073ADF03D
user.mercury={DES3S}9504D70960F68B9C
```

Using a text editor, change the password of the relevant user and save the file. (Note that when the tuning agent is next started, the password will be encrypted.)

Changing the Tuning Agent's Port

By default, all communication between the Console and the tuning agent is handled by a proprietary messaging protocol encoded and secured over Secure Sockets Layer (SSL). You can change the tuning agent's default port from OTP-SSL 4863 to a user defined port.

When you install the tuning agent, the installation process installs the pe_agent.bat (Windows) and pe_agent (UNIX) batch files on the host.

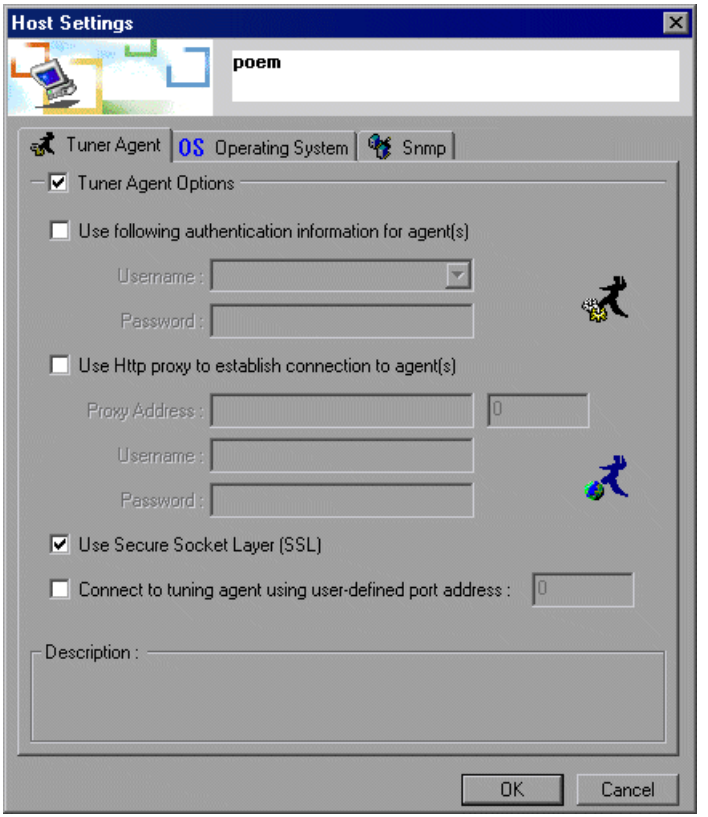
Tip: On a Windows host, the pe_agent.bat file is located in the *Program Files\Mercury Interactive\Performance Expert\agent\bin* directory.

You can modify the agent's listening port using the Host Properties dialog box or the command line.

Specifying the port using the Host Properties dialog box:

1 In the Server Configurations tree, click the host and then click the **View Host Properties** button. Alternatively, right-click the host and choose Properties.

The Host Settings dialog box is displayed:



2 Check the **Connect to tuning agent...** box and specify the port in the adjacent field.

Specifying the port in the command line:

➤ Specify the listening port when you invoke the batch file from the command line. Use the following syntax:

```
pe_agent <Port#> <SSL_Flag> <Path_to_PE_Installation>
```

The following examples illustrate how to use the batch file:

Command	Action
pe_agent 1234	Launches the tuning agent at port 1234 with SSL enabled.
pe_agent 1235 false	Launches the tuning agent at port 1235 with SSL disabled.
pe_agent 1236 true C:\LoadRunner Tuning Module	Launches the tuning agent at port 1236 with SSL enabled, using configuration and tuners from C:\LoadRunner Tuning Module.

➤ Set the tuning agent's environment variables and then run the batch file. The following example illustrates this method:

```
set PE_USE_PORT=4444
set PE_USE_SSL=true
set PE_HOME=C:\LoadRunner Tuning Module
pe_agent
```

Tip: On a UNIX host, use a similar procedure, depending on your UNIX shell.

Using the Performance Tuner Registry

The Performance Tuner Registry allows you to configure the tuners on a host.

Each environment (for example, IIS, Apache, and Oracle) has a dedicated tuner that is capable of administering the environment. In some cases the tuner needs information on where to find the application that needs tuning, and may also need logon credentials.

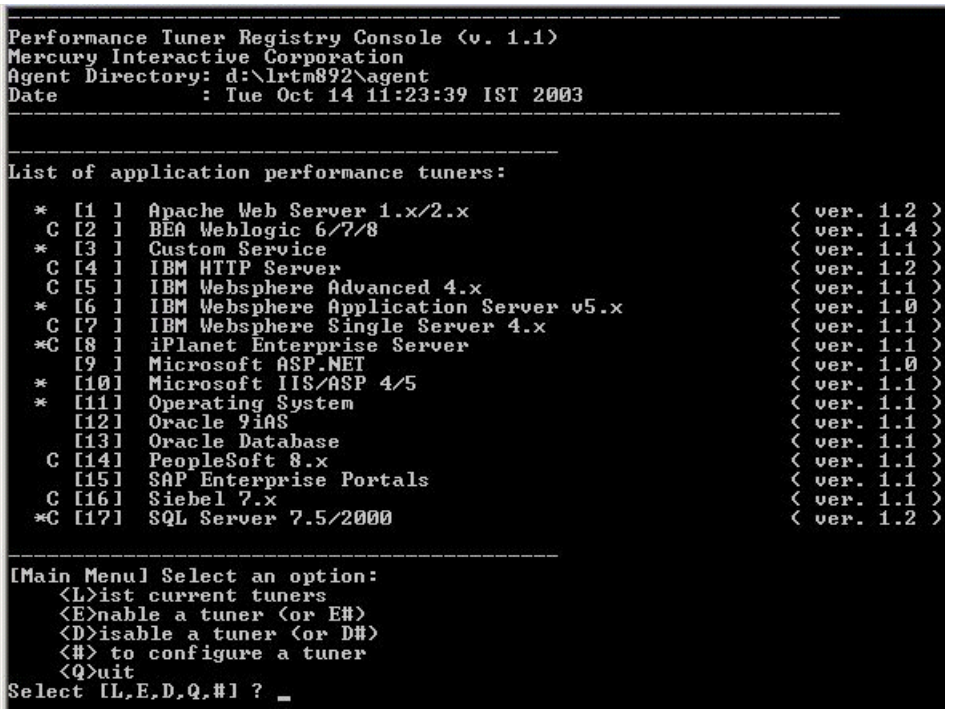
The Performance Tuner Registry provides a command-line interface for configuring the individual tuners.

To invoke the Performance Tuner Registry:

1. On the host that is being tuned, set the PE_HOME environment variable so it points at the Performance Tuner's home directory. **Note:** When you install the tuning agent remotely from the Console machine or from the installation CD, LoadRunner Tuning Module sets the environment variable to this value.
2. From the command line, enter one of the following commands:
   - pe_registry.bat (Windows)
   - pe_registry (UNIX)

Note: Alternatively, if you installed the tuning agent locally from the CD, you can invoke the Performance Tuner Registry by choosing **Start** > **Programs** > **Performance Expert** > **Tuning Agent Configuration**.

LoadRunner Tuning Module invokes the Performance Tuner Registry and displays the Performance Tuner Registry Console:



The window displays a list of all the services for which tuners are available. The services are marked as follows:

Sign	Indicates	Comments
*	The tuner is active on the host (the service is installed on the machine).	If the service is not installed on the host, the tuner is not marked as active.
C	You may need to configure the tuner before it can become active. For example, you may need to specify where the service is installed.	
X	The tuner is disabled.	The user cannot view information about the service for which the tuner is intended, and cannot tune or administer the service.

The Performance Tuner Registry Console allows you to perform the following actions:

- List the current tuners
- Enable a tuner
- Disable a tuner
- Configure a tuner
- Quit the Performance Tuner Registry Console

To list the current tuners:

- Type "L" and press <Enter>.

To enable a tuner:

1. Type "E" and press <Enter>.

   The Performance Tuner Registry Console asks you to enter the ID of the tuner that you want to enable.

2. Type the tuner ID (the number in brackets that appears before the tuner's name) and press <Enter>.

Note: If the tuner requires configuration (indicated by the letter "C" before the tuner's name), you cannot enable it.

The next time you list the current tuners, the enabled tuner appears with an asterisk.

To disable a tuner:

1 Type "D" and press <Enter>.

The Performance Tuner Registry Console asks you to enter the ID of the tuner that you want to disable.

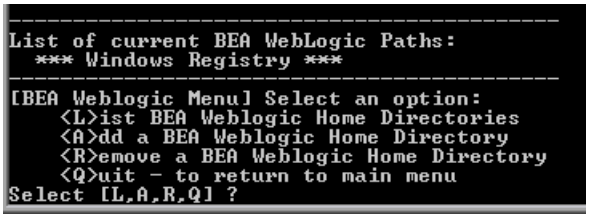
2 Type the tuner ID (the number in brackets that appears before the tuner's name) and press <Enter>.

The next time you list the current tuners, the enabled tuner appears without an asterisk.

To configure a tuner:

1 Type the tuner's number to select it, and press <Enter>.

The configuration menu for the selected tuner is displayed. The following example shows the configuration menu for the WebLogic Application Server tuner:



2 Follow the onscreen instructions for configuring the selected tuner.

To quit the Performance Tuner Registry Console:

➤ Type "Q" and press <Enter>.

Automatically Starting the Tuning Agent when Booting

You can configure the host to start the tuning agent automatically when the host is started.

On a Windows system: Using regedit.exe, create a key in the Windows registry under:
[*HKLM\Software\Microsoft\Windows\CurrentVersion\Run]*
PE_AGENT = "%PE_HOME%\agent\bin\pe_agent.bat" (REG_SZ)

On a UNIX system: Update your startup file under /etc, and add a command to launch pe_agent.

When you launch the tuning agent (pe_agent.bat or pe_agent), you can include the following optional arguments:

Argument	Specifies	Values
[PE_USE_PORT]	listening port	Default: 0
[PE_USE_SSL]	whether SSL is enabled	True False
[PE_HOME]	location of PE_HOME (if the PE_HOME environment variable is not defined)	

Following is the syntax for launching the tuning agent:

```
pe_agent[.bat] [PE_USE_PORT] [PE_USE_SSL] [PE_HOME]
```

Following is an example of how to run the pe_agent.bat command:

```
pe_agent.bat 0 true C:\Program Files\Mercury Interactive\PerfExpert
```

Starting and Stopping the Apache and IBM HTTP Servers

To enable the Console to remotely start and stop an Apache Web Server or IBM HTTP Server running on a host machine, you need to perform the following preparatory actions:

➤ Create scripts for starting and stopping the Web server, and place them in a directory on the host machine.

➤ On the host machine, configure the tuner. This includes specifying the path to the Web server and the scripts for starting and stopping it.

Creating Scripts for Starting and Stopping the Web Server

To enable the tuning agent to remotely start and stop the Apache Web server or IBM HTTP Server, you need to create scripts and place them in the relevant directory of the Web server installation.

1 Use a text editor to create the following batch files:

On a Windows system:

Web Server	Filename	Content
Apache	start.bat	Apache.exe -w -n "Apache" -k start
Apache	stop.bat	Apache.exe -w -n "Apache" -k stop
IBM HTTP Server	start.bat	net.exe start "IBM HTTP Server"
IBM HTTP Server	stop.bat	net.exe stop "IBM HTTP Server"

On a UNIX system:

Web Server	Filename	Content
Apache	start	apachectl -w -n "Apache" -k start
Apache	stop	apachectl -w -n "Apache" -k stop

To enable starting and stopping an IBM HTTP Server running on a UNIX host machine, define shell scripts (start and stop) for starting and stopping the IBM HTTPD service.

2 Place the batch files in the relevant directory:

- **For Apache Web Server**: In the \bin directory of the Apache installation. For example, on a Windows computer, this might be located at C:\Program Files\Apache Group\Apache\bin. On a Linux system, you need to create the /bin directory (with the start and stop scripts) under /etc/httpd.
- **For IBM HTTP Server**: In the directory containing the Apache.exe file. This is typically the **IBM HTTP Server** directory.

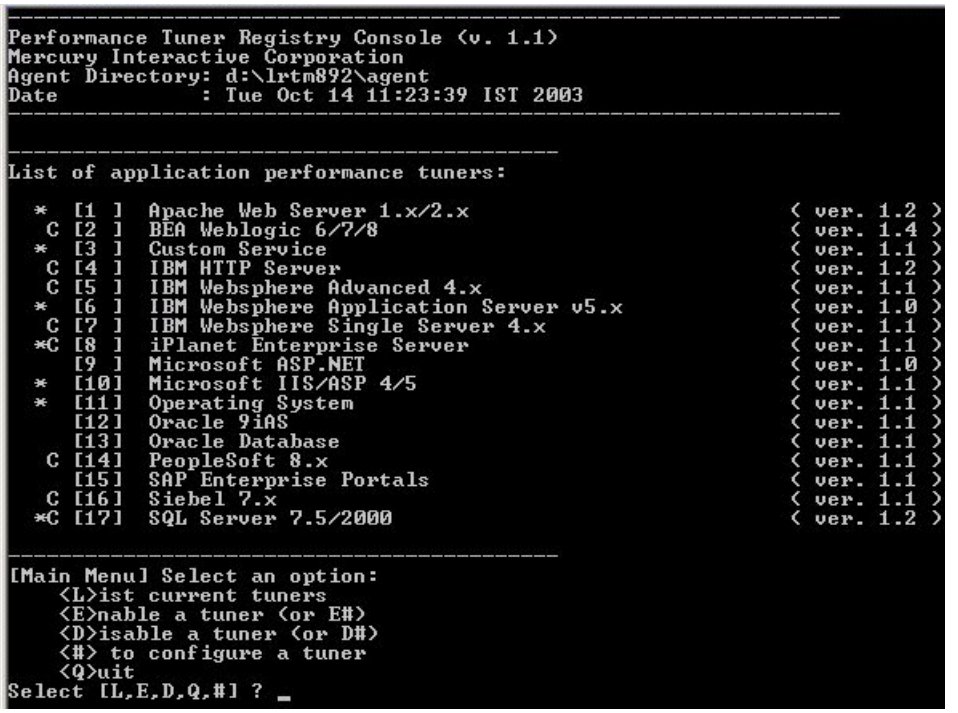
Configuring the Tuner

You use the pe_registry batch file to configure the tuner, specifying the path to the Web server and the start and stop scripts on the host machine.

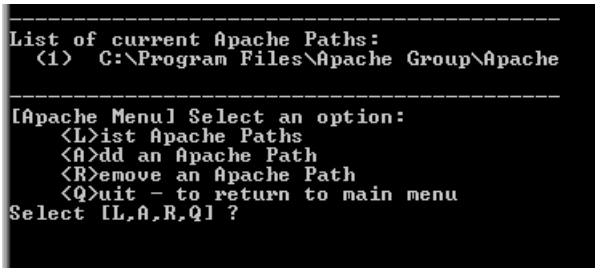
To specify the path to the Web server:

1 On the host machine, invoke the Performance Tuner Registry Console (see "Using the Performance Tuner Registry," on page 704).

The list of tuners is displayed.



2 Type the number of the Web server tuner and press <Enter>. (In the example above, you would type the number 1 for the Apache Web Server, or 3 for the IBM HTTP Server.) This displays the tuner's configuration menu. The current paths to the Web server are listed above the menu.



The path must specify the directory that contains the Apache.exe file, and must **not** specify a subdirectory of that directory. For example, for an

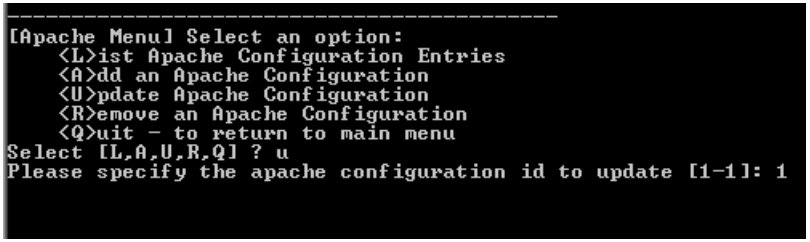
Apache Web server installation on a Windows computer, C:\Program Files\Apache Group\Apache is acceptable; C:\Program Files\Apache Group\Apache**\bin** is not.

On a Linux computer (for the default Apache installation), the path should be /etc/httpd.

3 If the correct path is not listed, you need to add it to the path list. To add the path to the list, type **A** and press <Enter> to choose the **Add an Apache Path** or **Add an IBM HTTP Server Path** option, specify the path to the directory that contains the Apache.exe file, and press <Enter>. The path to the Web server is added and displayed by the Performance Tuner Registry.

To specify the starting and stopping scripts:

1 From the Web server configuration menu, type **U** to update the configuration and press <Enter>. The Performance Tuner Registry asks for the ID of the Web server that you are configuring. (This is the number to the left of the relevant path in the list of current Apache paths.)



2 Type the relevant ID and press <Enter>.

The application displays the path to the Web server's home directory. This is the path that you specified in Step 2 on page 711.

If the displayed path is incorrect, enter the correct one.

3 Press <Enter>. The application displays the Web server's version number.

If the version is incorrect, enter the correct one.

4 Press <Enter>. The application displays the Web server's startup script (if one has been specified).

5. Ensure that the displayed script has the name that you specified above in "Creating Scripts for Starting and Stopping the Web Server," on page 709.

   If the name that you defined is not displayed, enter the correct name (start.bat for a Windows system, or start for UNIX).

6. Press <Enter>. The application displays the Web server's shutdown script (if one has been specified).

   Ensure that the displayed script has the name that you specified above in "Creating Scripts for Starting and Stopping the Web Server," on page 709.

7. If the name that you defined is not displayed, enter the correct name (stop.bat for a Windows system, or stop for UNIX).

8. Press <Enter>.

   The Web server's configuration menu is displayed.

9. Type **Q** and press <Enter> to exit the configuration menu. The Performance Tuner Registry main menu is displayed.

10. Type **Q** and press <Enter> to exit and save your settings.

48

Configuration Tab Functions

This chapter describes various functions that are available to you via the Configuration tab.

The chapter describes the following functions:

- Start a Service
- Stop a Service
- Reboot Host Machines
- Reconnect the Console to a Server
- Stop the Tuning Agent
- Print Host Configurations
- Refresh Configuration of Service/Host
- Remove Host from Server Configurations Tree

About Configuration Tab Functions

Some tuning functions can be performed by all categories of users—whether they have only read-only permissions, or update permissions, or full administrative privileges. Some functions are available only to users who have administrator privileges (that is, users who connect with the *admin* username). If you attempt to perform a function for which you are not authorized, LoadRunner Tuning Module displays an error message.

For details of the various types of users and their usernames and passwords, see the table on page 665.

Start a Service

(For users with administrator access only).

To start a Windows service:

1. Expand the host computer's Windows Services element. The host's services are displayed.
2. Right-click the service that you want to start, and choose Admin > Start Service.
3. To verify that the service has started, right-click the service and choose Refresh. If the service has started, its status (in the right side of the Information tab) changes to **Running**.

To start other types of services:

1. Right-click the service, and choose Admin > Start Service.

   LoadRunner Tuning Module displays a dialog box requesting confirmation.
2. Click **Yes** to start the service.

Note: After you choose Start Service, you may have to wait for a while until the service has started.

Stop a Service

(For users with administrator access only).

To stop a Windows service:

1. Expand the host computer's Windows Services element. The host's services are displayed.
2. Right-click the service that you want to start, and choose Admin > Stop Service.

3 To verify that the service has stopped, right-click the service and choose Refresh. If the service has stopped, its status (in the right side of the Information tab) changes to **Stopped**.

To stop other types of services:

1 Right-click the service, choose Admin, and then choose Stop Service.

LoadRunner Tuning Module displays a dialog box requesting confirmation.

2 Click **Yes** to stop the service.

Note: After you choose Stop Service, you may have to wait for a while until the service has stopped.

Reboot Host Machines

(For users with administrator access only).

LoadRunner Tuning Module allows you to reboot host machines from the Console machine.

If the host machine is running Windows, you need to specify its username and password to enable the host to log in automatically after rebooting. You specify the username and password via the Performance Tuner Registry.

To specify the host machine's username and password:

1 Invoke the Performance Tuner Registry. (See "Using the Performance Tuner Registry," on page 704 for details.)

2 Type **11** (for the Operating System tuner) and press <Enter>. The OS Menu is displayed.

3 Type **C** and press <Enter>. The Performance Tuner Registry prompts you to enter the username and password.

4 Enter the username and password and exit the Performance Tuner Registry.

To reboot a host machine:

1 Right-click the server's icon, choose Admin, and them choose Reboot Host.

LoadRunner Tuning Module displays a dialog box requesting confirmation.

2 Click **Reboot** to reboot the host machine.

To reboot all the host machines:

1 Right-click the Server Configurations node, choose Admin, and then choose Reboot All.

LoadRunner Tuning Module displays a dialog box requesting confirmation.

2 Click **Reboot All** to reboot all the host machines.

Reconnect the Console to a Server

Once the tuning agent has been installed on a server, you can access the server by clicking the **Connect to Host button** on the toolbar. In the Connect to Server dialog box, choose the server you want to tune and click **Connect**. If the tuning agent is running on the server, the Console connects to the server via the tuning agent, and shows you the server information.

If the tuning agent has been installed on the server but is not currently running, start it by clicking the **Start Tuning Agent** button. Clear the Auto-Install... box, verify that the other fields have the correct values, and click **Start**.

On the Console machine, the server icon in the Server Configurations tree changes to blue, indicating that the connection to the server is alive.

Stop the Tuning Agent

To stop the tuning agent that is running on the server, click the server's icon and click the **Stop Tuning Agent** button. When LoadRunner Tuning Module asks you to confirm the action, click **Yes**.

Print Host Configurations

You may find it useful to keep a hard-copy record of a host's tuning settings. LoadRunner Tuning Module allows you to print the configuration settings as they appear in the Configuration tab.

To print the configuration settings of a host or service:

Select the host or service, click the **Generate HTML report** button. The HTML report is displayed by your browser. Print the report from the browser.

Refresh Configuration of Service/Host

➤ To reload the current settings from a host machine, click the host's icon in the Servers Configurations tree and click the **Refresh Configuration of Service/Host** button. Alternatively, right-click the host's icon and choose Refresh.

Remove Host from Server Configurations Tree

To remove a host from the Server Configurations tree, right-click the host's icon and choose Remove.

49

Tuning UNIX Hosts

This chapter describes the permissions, access rights and actions that you need to perform before you can tune a host machine running a UNIX operating system.

It includes the following sections:

- Using Telnet
- Redirecting Script Output
- Solaris Requirements
- IBM AIX Requirements
- HP-UX Requirements
- Linux Requirements

Using Telnet

In some secure environments, it is recommended to use Telnet to install, start, or stop a tuning agent on a UNIX host.

To launch Telnet, click **Launch Terminal Client**, and follow the installation instructions in “Local Installation of a Tuning Agent” on page 669.

Redirecting Script Output

It is recommended to redirect the output of the start and stop scripts to a user-defined log file, especially if the scripts print information to the

standard output device. This is because some versions of JRE may prevent proper execution of scripts that produce a lot of output.

To redirect the script output, use one of the following methods:

➤ Specify the script path with redirection to the log file, as in the following example:

```
mystartup.sh > startup.log
```

➤ Update the script code to automatically generate the log file.

Solaris Requirements

Access Rights and Permissions

1 The user running the tuning agent must have access rights and permissions to execute the following commands:

Command	Gives this information	Default location
psrinfo	CPU speed and number of CPUs	/usr/sbin
prtconf	Total RAM	/usr/sbin
vmstat	Available RAM	/usr/sbin
swap	Total and available virtual memory	/usr/sbin
ndd -- [program]	Network tuning parameters	/usr/sbin
/etc/system -- [file]	File system tuning parameters	

PATH Environment

Update the PATH environment so it includes the /usr/sbin directory.

Verification

After updating the path, execute each of the commands listed in the table above to verify that you have the appropriate access rights for executing them. **Note:** Some commands may require root privileges. Also verify the existence of the system file in the /etc directory.

IBM AIX Requirements

Access Rights and Permissions

1 The user running the tuning agent must have access rights and permissions to execute the following commands:

Command	Gets this information	Default location
uname	Name and version of the operating system	/usr/bin
bootinfo	Total RAM	/usr/bin
vmstat	Total and available virtual memory	/usr/bin
lsdev	Number of processors	/usr/bin
/usr/samples/kernel/vmtune -- [program]	File system tuning parameters	
no -- [program]	Network tuning parameters	

2 If the user's Java installation does not support SSL with RSA encryption, the user can launch the tuning agent without using SSL. To do this, launch the tuning agent with the SSL flag set to False, as in the following example:

```
pe_agent 4862 false
```

Note: The default port for non-SSL connections is 4862.

When connecting to a tuning agent whose port and SSL state have non-default settings, the client user should update the host properties for the target host appropriately.

PATH Environment

Update the PATH environment so it includes the /usr/bin directory.

Verification

After updating the path, execute each of the commands listed in the table above to verify that you have the appropriate access rights for executing them. **Note:** Some commands may require root privileges.

HP-UX Requirements

Access Rights and Permissions

The user running the tuning agent must have access rights and permissions to execute the following commands:

Command	Gets this information	Default location
uname	Name and version of the operating system	/usr/bin
model	CPU speed	/usr/sbin
swapinfo	Total and available virtual memory	/usr/sbin
dmesg	Total and available RAM	/etc
ioscan	Number of processors	/usr/sbin
ndd -- [program]	Network tuning parameters	

PATH Environment

Update the PATH environment so it includes the following directories:

- /usr/sbin
- /usr/bin

➤ /etc

Verification

After updating the path, execute each of the commands listed in the table above to verify that you have the appropriate access rights for executing them. **Note:** Some commands may require root privileges.

Linux Requirements

Access Rights and Permissions

The user running the tuning agent must have access rights and permissions to execute the following commands:

Command	Gets this information	Default location
uname	Name of the operating system	/usr/bin
/proc/cpuinfo file	Operating system version, CPU speed and total number of CPUs	
/proc/meminfo file	All memory statistics	
sysctl -- [program]	File system and network tuning parameters	

PATH Environment

Update the PATH environment so it includes the /usr/bin directory.

Verification

After updating the path, execute each of the commands listed in the table above to verify that you have the appropriate access rights for executing them. **Note:** Some commands may require root privileges. Also check for the existence of the meminfo and cpuinfo files in the /proc directory.

Part VII

Appendixes

A

Analysis Graphs

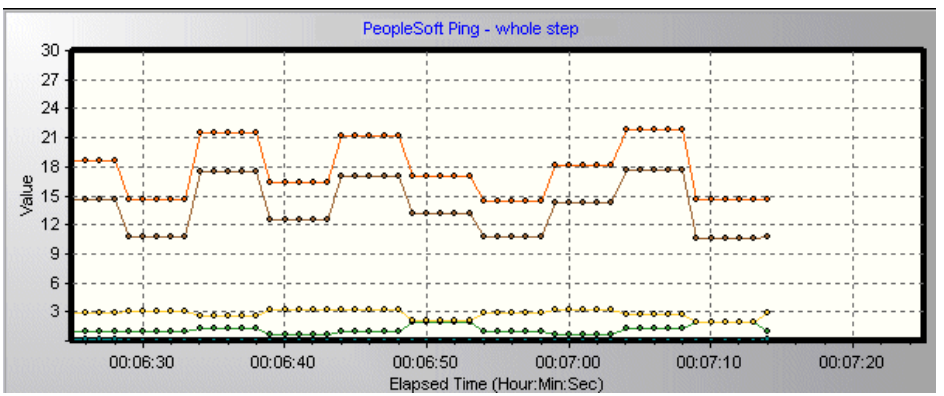
In addition to the LoadRunner Analysis graphs, the LoadRunner Tuning Module provides the following graphs:

- PeopleSoft Ping Graph
- Distributed Denial of Service Graph
- F5 BIG-IP Graph

PeopleSoft Ping Graph

The PeopleSoft Ping graph shows the resource usage of your PeopleSoft 8.42 server as a function of the elapsed session step time.

The x-axis represents the elapsed time. The y-axis represents the resource usage.

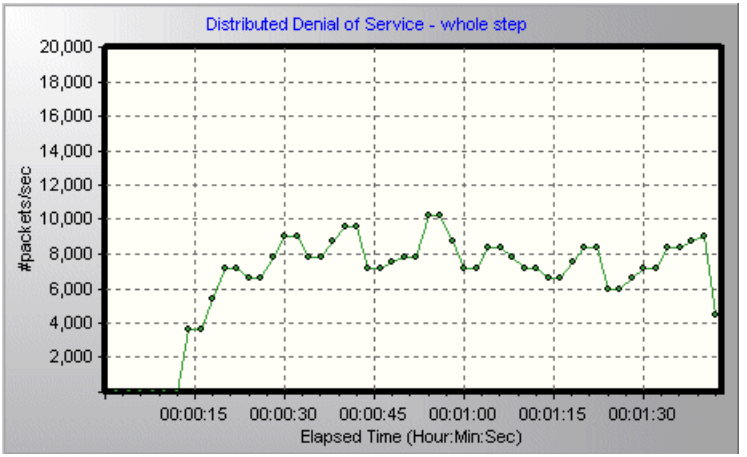


Note: To obtain data for this graph, you need to enable the PeopleSoft Ping monitor (from the Console) and select the measurements you want to display, before running the session step.

Distributed Denial of Service Graph

The Distributed Denial of Service graph displays the number of packets per second sent to the specified target to cause denial of service.

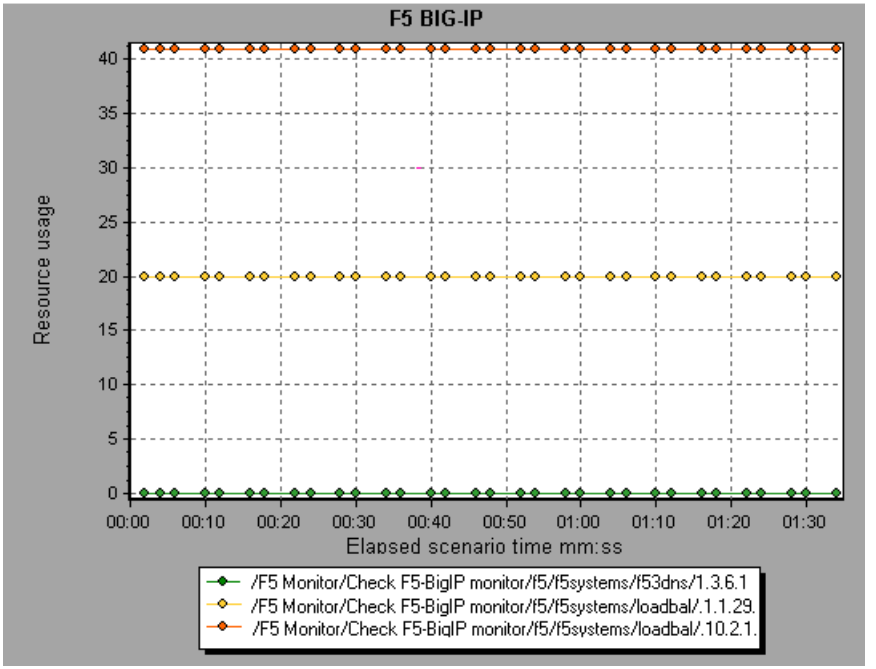
The x-axis represents the elapsed time. The y-axis represents the number of packets sent per second.



For more information on activating and configuring the Security monitor, refer to the *LoadRunner Controller User's Guide*.

F5 BIG-IP Graph

The F5 BIG-IP graph provides information about the content of event logs and other data from the F5 BIG-IP load balancing device. The x-axis represents the elapsed time. The y-axis represents the resource usage.



The F5 BIG-IP graph provides the following measurements:

pktsin
pksout
concur
portdeny
uptime
droppedin
droppedout
MemoryUsed

B

Troubleshooting the Console

LoadRunner Tuning Module enables you to test entire applications. If one of the components of the application is not configured properly, LoadRunner Tuning Module sessions will not run.

This appendix discusses the most common LoadRunner Tuning Module problems:

- LoadRunner Tuning Module Communications
- Failure to Communicate with a Load Generator
- Failure to Connect to the AUT Database
- Failure to Access Files
- Failed Vusers or Transactions
- Increasing the Number of Vusers on a Windows Machine
- Running DDoS Scripts on LoadRunner Load Generators
- Troubleshooting Firewalls
- Troubleshooting Remote Tuning

About Troubleshooting

LoadRunner Tuning Module relies heavily upon communication between machines on a network. If communication is not established properly, the Console will be unable to send commands to remote load generators and the session will fail. By understanding the reason for the failure and determining when the failure occurred, you can solve most of the communication-related problems.

In order to ensure that the problem lies with your session and not your script, you should verify that your script runs properly on all remote load generators as a stand-alone:

- Test your GUI scripts on Windows platforms using WinRunner.
- Test your scripts on UNIX platforms by running them from the command line.
- Test all other types of scripts on Windows platforms by running them from VuGen, or by running a single user from the Console.

Note: When a test runs in VuGen, the full browser is used. This differs from a test run in the Console, where only the browser basics are used. There may be occasions when a test passes its run in VuGen, but fails when it is run in the Console. Before running a session in the Console with multiple Vusers, run a single Vuser to ensure the test is bug free.

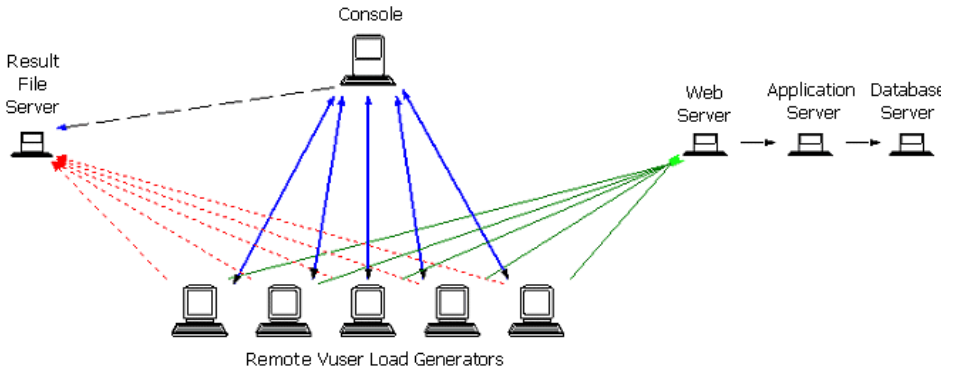
For more information on running scripts in stand-alone mode, refer to the appropriate guide for creating scripts.

LoadRunner Tuning Module Communications

Most communication problems can be solved if you understand your LoadRunner Tuning Module configuration. This knowledge helps you to determine the source of the problem and perform the necessary actions to correct it.

The following diagram illustrates a sample network running LoadRunner Tuning Module. There are five servers: The LoadRunner Tuning Module Console, the Web server, the application server, the database server, and the file server which stores the session results (note that result files can also be saved on a non-dedicated server). There are five remote load generators, each one running multiple Vusers.

The arrows indicate the type of communication necessary between the elements of the network. The Vusers communicate with the Console in both directions (send/receive), but with the file server in one direction (send). The Console must have access to the file server. All Vusers participating in the session must be able to communicate with the Web server in both directions (send/receive). In order for a client machine to connect to the server machine, it must be able to resolve the server machine name.



If any of the connections are broken, the session will fail.

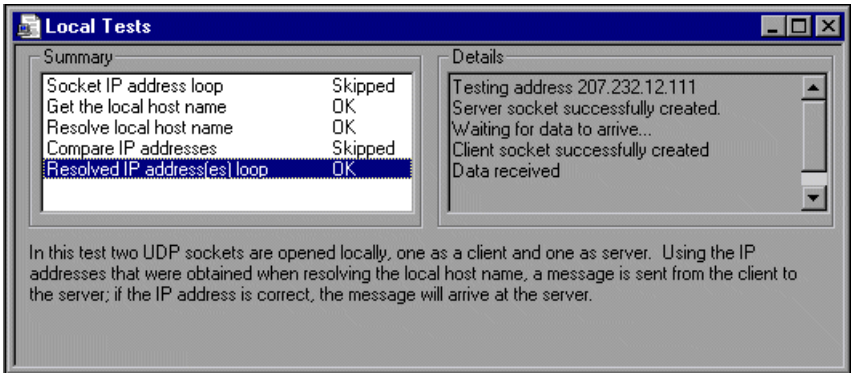
Failure to Communicate with a Load Generator

The most common communication error is the failure of the Console machine to connect with a remote load generator. Check the following items:

- TCP/IP setup
- TCP/IP connectivity
- Load generator connections
- UNIX shell

Checking TCP/IP Setup

The first step in checking your configuration is to verify your machine's TCP/IP setup. LoadRunner Tuning Module includes a utility called Hostinfo (hostinfo.exe), located under LoadRunner Tuning Module's bin directory. This utility provides information about the current machine—local name and local address. It also insures that TCP/IP is properly installed on the current machine.



When you invoke Hostinfo, it automatically verifies the TCP stack by:

- retrieving and resolving the local machine name
- retrieving and resolving the IP address

To resolve the IP address, Hostinfo tries to communicate using two UDP sockets on the same machine. It verifies that the IP address obtained while resolving the machine name is the same as the actual IP address of this machine.

To display the results of a test in the Details box, highlight the test name.

Note that the Edit menu in Hostinfo allows you to copy all machine information to the clipboard for sending to support personnel.

Checking TCP/IP Connectivity

Make sure that TCP/IP connectivity is functional on the Console and Vuser machines. Use a ping utility or type ping <server_name> from the DOS command line to verify communication with a remote machine. Make sure that the remote load generator and Console machines can ping each other by IP addresses and machine names.

If the ping does not respond, or fails with a timeout, then the machine name is not recognized. To solve this problem, edit the hosts file, located in the *WINNT\system32\drivers\etc* directory, and add a line with both the IP address, and the name. For example:

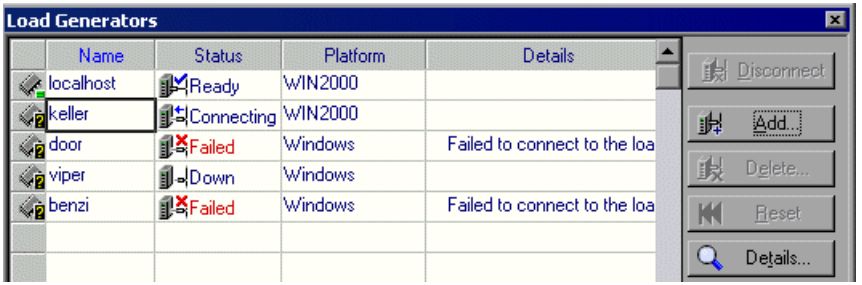
```
#       102.54.94.97     rhino.acme.com      # source server
#       38.25.63.10      x.acme.com          # x client host
```

Load Generator Connections

To verify the load generator connectivity, connect to each one of the remote load generators from the Console's Load Generators dialog box. In the load generator's Platform field, select a Windows or UNIX platform. Select the load generator(s) and click the Connect button. The status changes to *Connecting*.

If the Connection fails, the status changes to *Failed* and details are written to the Details box. Double-click the details box for more information about a failure.

If a connection succeeds, the status changes to *Ready*, and the actual platform name appears in the Platform box (such as WINNT, UNIX, etc.)



If your session uses several domains (for example, Vusers on a different domain than the Console), the Console may have trouble communicating with the load generators. This occurs because the Console uses the short load generator name—not including the domain—by default. To solve this, you must tell the Console to determine the full load generator names, including the domains.

Modify the *miccomm.ini* file in the Console machine's Windows directory as follows:

```
[tcpnet]
LocalHostNameType= 1
```

The possible values for LocalHostNameType are:

0 - Attempt to use the full machine name.
1 - Use the short machine name. This is the default.

Note: In certain environments such as WINS, load generators are unable to resolve machine names.

Connecting to a Console with Multiple IP Addresses

If the load generator machine does not recognize the Console machine by its short name or full name, and the Console machine has more than one IP address, you can define an alias name for the Console machine in the load generator's *hosts* file, located in the WINNT\system32\drivers\etc directory. The alias name should point to the IP address you want the load generator to recognize. For example: 255.0.0.1 delta.

UNIX Shell

For UNIX Vusers, make sure that the Windows Console can execute a remote shell command. Type the following at the DOS command prompt: rsh -l <UNIX user login name> <load generator name> <command>. If you get a message indicating that permission is denied, make sure the *.rhosts* file in your UNIX home directory contains Console machine permission for the user login name. In some cases, a "+" character must be added at the end of the *.rhosts* file. For example, if you log on to the Console as *bill* and connect

to the UNIX load generator as *mike*, you must ensure that *mike* allows *bill* to log on using his name. This can be done by adding the line "+ bill" at the beginning of mike's *.rhosts* file.

For more information on setting user login names, see "Configuring Load Generator Settings" on page 107.

To use UNIX without RSH:

1 On the UNIX Load Generator machine, run the agent daemon by running the following command from *<LoadRunner Tuning Module directory>/bin*:

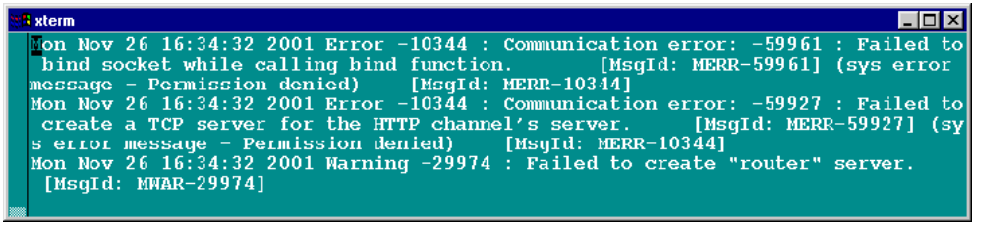
```
m_daemon_setup -install
```

This runs a daemon called m_agent_daemon, and if successful you will receive a message: m_agent_daemon installed successfully.

The agent will now keep running, even if the user is logged off. It will only stop running using the command explained in step 3, or by rebooting the machine.

- If you receive the message ERROR: File m_agent_daemon doesn't exist, this means that you are not in the same directory as the file (meaning not in *<LoadRunner Tuning Module directory>/bin* directory, or the file really doesn't exist, which indicates a problem with the installation).
- If a daemon of this name is already being run by the same user you will receive the following warning:
  WARNING: Could not install m_agent_daemon, reason - user <user_name> is already running m_agent_daemon on this machine.
- If an error occurred, you will receive the following error message:
  ERROR: Could not install m_agent_daemon. Check log file m_agent_daemon[xxx].log in your temp directory.

➤ If you look at the log file m_agent_daemon[xxx].log in the temp directory, you will see the following errors, even if the installation succeeded:



These messages appear because the LoadRunner Tuning Module agent always tries to open port number 443 (because any agent can be a MI Listener, and the MI Listener always listens to this port), and in UNIX machines, this port cannot be opened by any user except for the root user. However, this will not interfere with using this agent for the Load Generator machine.

2 In the Console, in the **Generators** > **Load Generator Information** > **Unix Environment** tab, check the **Don't use RSH** option. Then connect as usual.

3 To stop the agent daemon, run the following command the *<LoadRunner Tuning Module directory>/bin* directory: m_daemon_setup -remove

This stops the m_agent_daemon, and if successful you will receive a message: m_agent_daemon removed successfully.

➤ If no daemon of this name is being run by this user, you will receive the following warning:
WARNING: Could not remove m_agent_daemon, reason - user <user_name> is not running m_agent_daemon on this machine.

➤ If an error occurred, you will receive the following error message:
ERROR: Could not remove m_agent_daemon. Check log file m_agent_daemon[xxx].log in your temp directory.

Failure to Connect to the AUT Database

If you are running a database application, you must ensure that all remote clients can connect with the database server. If network or configuration errors occur when the client accesses the server, you must correct them before running a session. To ensure that your client application can connect with the database server, perform the following tests.

- Ping
- SQL utilities

Ping: Ensure that the client can communicate with the database server using TCP/IP. Use a ping utility or type ping <server_name> from the DOS command line.

SQL Utilities: Use a simple utility such as ISQL or SQLPLUS to log on to the database server and perform several basic operations.

Failure to Access Files

A LoadRunner Tuning Module session will fail if the result path or script is inaccessible to one or more of the participating machines. Check the following items:

- Path Translation
- Script
- Result Path

Path Translation: A script's location (path) is always based on the Console machine's mapping of that location. If a Vuser load generator maps to the script's path using a different name, path translation is required. Path translation translates the Console's mapping of a given location to the Vuser load generator's mapping. For example, if one machine maps the script directory as *g:\test*, while another maps it as *h:\test*, the paths should be translated.

Path translation is also effective across platforms—between Windows and UNIX. You use path translation to translate the Windows Console paths into paths recognized by UNIX.

Note: Path translation is only required if you chose to save all scripts and results to a shared network drive. In the default setup, LoadRunner Tuning Module saves files locally and collates them to the Console machine; no path translation is required.

Suppose that your script is in the /usr/jon/lr_test1 directory and runs on the UNIX machine, *sunny*. To translate it from the Windows Console machine, *pc1*, where your UNIX directory is mapped as *r*, enter the following line in the path translation table:

pc1	r:\	/usr/jon	sunny

To translate the f:\qa Console directory to all load generator machines running */m/qa/lr_test2/lr_test2.usr* on a UNIX platform, type:

win	f:\qa	/m/qa	UNIX

If the paths are not translated properly, the session will fail. For more information about path translation, see Appendix D, "Performing Path Translation."

Script: Make sure that the script is accessible to all load generators participating in the session through path translation and permissions. View or run the script as a stand-alone on each of the participating load generators.

Result Path: Make sure that the result path is accessible to all load generators participating in the session through path translation and permissions. Check the permissions of the result directory files and modify them if necessary.

Failed Vusers or Transactions

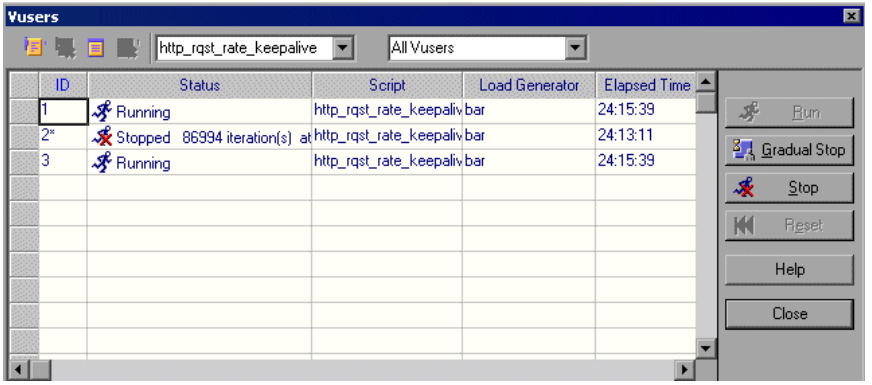
LoadRunner Tuning Module Vusers or transactions may fail for a variety of reasons relating to the network, database, or actual script. You can find information about session runs from the following sources:

- Run View
- Output Window
- Output File (excluding GUI Vusers)
- Analysis Reports and Graphs

Run View

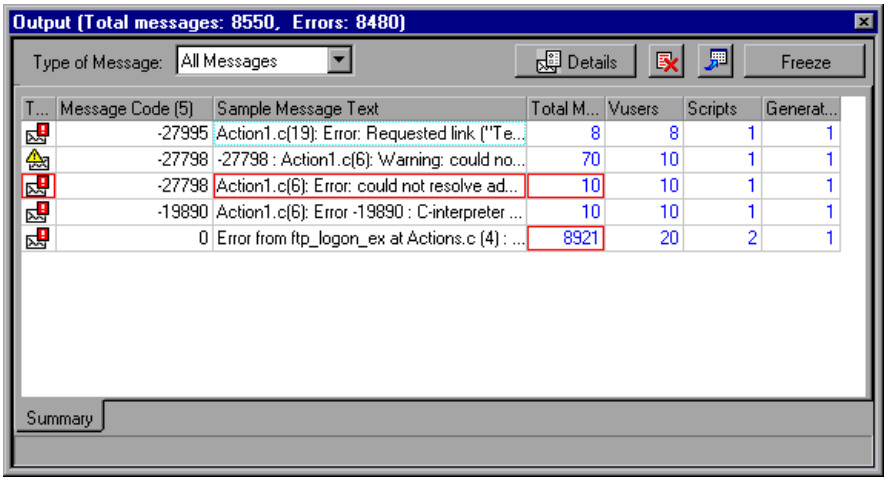
The Run view is part of the LoadRunner Tuning Module Console. The Session Groups window in the top left-hand corner of the view indicates the status of the Vuser groups during and after the session run. During the session run, the columns will show a PENDING, INITIALIZING, READY, RUNNING, or RENDEZVOUS status. You can also view the status of each individual Vuser in the Vusers dialog box. If a Vuser fails and does not complete the script execution, LoadRunner Tuning Module displays an error status. If a Vuser completes the script execution, LoadRunner Tuning Module indicates the transaction status of a completed script run using the DONE.FAILED or DONE.PASSED status.

For more information about the Vuser states, see Chapter 17, "Running an Execution Step."



Output Window

View the Output window from the Console. The output window contains useful information for debugging a session. The output window lists five types of messages: errors, warnings, notifications, debug, and batch. An error message usually results in a failed script. A warning message indicates that the Vuser encountered a problem, but test execution continued. A notification provides useful information such as recorded think time values and other run-time information. A debug message is sent if you enable the debugging feature in **Tools > Options > Debug Information** (Expert Mode). Batch messages are sent instead of message boxes appearing in the Console, if you are using automation.



For more information about the Output window, see Chapter 18, "Viewing Vusers During Execution."

Output File

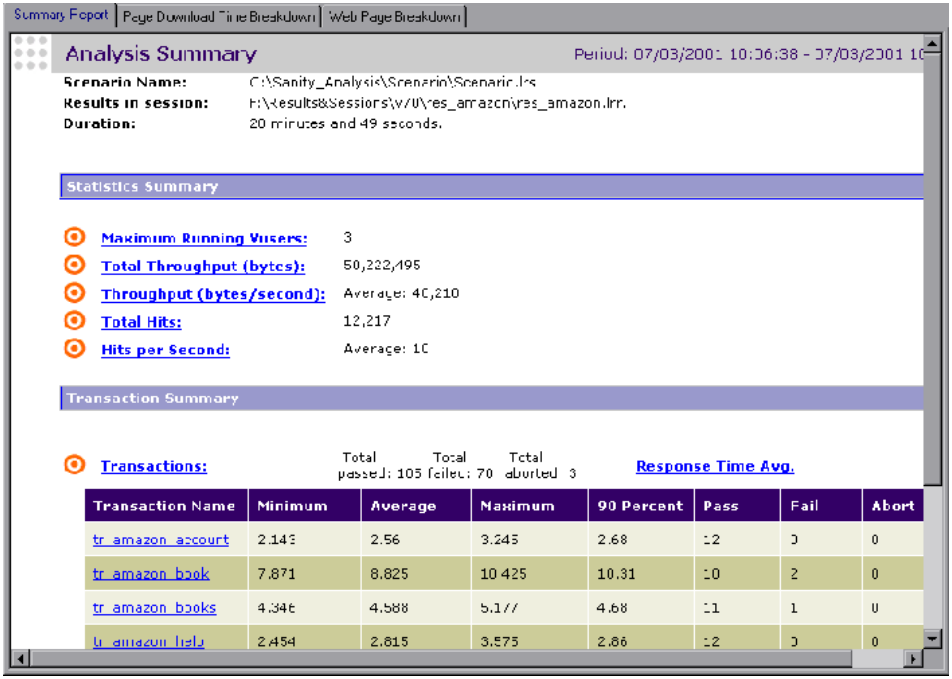
You can view information about script execution in an output file located in the Vuser result directory. The output file, *output.txt*, contains:

- a list of the primary functions called during the session
- error messages from the database server
- transactions and rendezvous information

The extent of the information sent to the output file depends on the output file settings. In the VuGen's run-time settings, you specify a Brief or Extended log. For the Extended log, you can specify a full trace, returned data, or current parameter value. An extended log is helpful for debugging a script, but if you are not debugging, Extended log is not recommended as it introduces extra overhead. For more information about configuring run-time settings, refer to the *Creating Vuser Scripts* guide.

Analysis Reports and Graphs

You can generate graphs and reports to view information about the session run. For example, the Session Summary report displays a table containing the session's run-time data and provides links to the following graphs: Running Vusers, Throughput (Web), Hits Per Second (Web), HTTP Responses per Second, Transaction Summary, and Average Transaction Response Time.



For more information on the available graphs and reports, refer to the *Analysis User's Guide*.

Increasing the Number of Vusers on a Windows Machine

Under the normal settings of a Windows machine, you are limited to several hundred Vusers. This limitation is related to the operating system and not to the CPU or memory.

To work around the limitation of the Windows operating system, modify the Windows Kernel as follows:

1. Save a copy of the registry file in case you have trouble with these modifications.
2. Run Regedit.
3. Go to following key in KEY_LOCAL_MACHINE:

   System\CurrentControlSet\Control\Session Manager\SubSystems

4. Select the Windows key. The default Windows key for NT 4.0 looks like this:

   ```
   %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows
   SharedSection=1024,3072
   Windows=On SubSystemType=Windows ServerDll=basesrv,1
   ServerDll=winsrv:UserServerDllInitialization,3
   ServerDll=winsrv:ConServerDllInitialization,2
   ProfileControl=Off MaxRequestThreads=16
   ```

   The SharedSection=1024,3072 key has the format xxxx,yyyy where:

   xxxx defines the maximum size of the system-wide heap (in kilobytes)

   yyyy defines the size of the per desktop heap.

5. Increase the SharedSection parameter by changing the yyyy settings from 3072 to 8192 (which is 8 MB).

   This setup successfully allowed 1250 Oracle Vusers to run on a Windows machine using 2 Pentium PRO 200 MHz with 1 GB RAM.

   Each Vuser in this setup used approximately 2MB memory. Other Vusers may require more memory.

LoadRunner Tuning Module was able to load over 2500 Vusers when the Windows terminal server was run as the Operating System and the above registry setting was changed.

The above registry changes enable you to run more threads, allowing you to run more Vusers on the machine. This implies that you are not bound by the Windows operating system; you are only bound by hardware and internal scalability limitations.

Running DDoS Scripts on LoadRunner Load Generators

Before using a LoadRunner load generator (as opposed to a LoadRunner Tuning Module load generator) to execute a DDoS script, you need to do the following:

1. Copy the ddos.dll file (located in the Tuning Module's bin directory on the Console machine) to the <load generator root>\bin directory on the load generator machine.
2. Add a DDOS section to the load generator's dat\mdrv.dat file. The section should read as follows:

```
[DDOS]

ExtPriorityType=protocol

WINNT_EXT_LIBS=ddos.dll

WIN95_EXT_LIBS=ddos.dll

LINUX_EXT_LIBS=

SOLARIS_EXT_LIBS=

HPUX_EXT_LIBS=

AIX_EXT_LIBS=

LibCfgFunc=extension_configure

UtilityExt=lrun_api

SecurityMode=On
```

Troubleshooting Firewalls

There are three log files which provide additional information about activity over the firewall.

The LoadRunner Tuning Module **agent log file** contains information about communication activity between the LoadRunner Tuning Module agent and the MI Listener.

➤ To open the file on Windows machines, right-click the LoadRunner Tuning Module agent icon in the system tray of the LoadRunner Tuning Module agent machine, and select **View Log**. Alternatively, open the latest *<temp_directory>\<LoadRunner_Tuning_Module_agent_startup<unique identifier>.log* file (if the LoadRunner Tuning Module agent is a process), or *<temp_directory>\LoadRunner_Tuning_Module_agent_service<unique identifier>.log* file (if the LoadRunner Tuning Module agent is a service), in a text editor.

➤ To open the file on UNIX machines, open the *<temp_directory>/m_agent_daemon<unique identifier>.log* file in a text editor.

➤ To increase the logging level, select Agent Settings from **Start->Programs->LoadRunner Tuning Module->Advanced Settings** (or open file *<LoadRunner_Tuning_Module_root>\launch_service\dat\br_lnch_server.cfg* in a text editor), and in the Log section, set AgentExtended to 1.

The **MI Listener log file** contains information about MI Listener communication with the LoadRunner Tuning Module agent and the Console.

To open the file, right-click the MI Listener Agent icon in the system tray of the MI Listener machine, and select **View Log**. Alternatively, open the latest *<temp_directory>\LoadRunner_Tuning_Module_agent_startup<unique identifier>.log* file (if the LoadRunner Tuning Module agent is a process), or *<temp_directory>\LoadRunner_Tuning_Module_agent_service<unique identifier>.log* file (if the LoadRunner Tuning Module agent is a service), in a text editor.

To increase the logging level, select **Start** > **Programs** > **LoadRunner Tuning Module** > **Advanced Settings** > **Agent Settings**, or open the *<LoadRunner_Tuning_Module_root>\launch_service\dat\br_lnch_server.cfg* file in a text editor. In the Log section, set AgentExtended to 1.

The **Console log file** contains information about communication activity between the Console and the MI Listener.

To open the file on Windows machines, open the *<temp_directory>\drv_log.txt* file in a text editor.

Verifying Connection Between LoadRunner Tuning Module Agent and MI Listener

If there is a proper connection between the LoadRunner Tuning Module agent and the MI Listener:

- On Windows platforms, the agent icon's light in the system tray will turn from red to green.
- On UNIX platforms, a file called *<Local_machine_key>_connected_to_MI_Listener* will be created in the temporary directory of the LoadRunner Tuning Module agent machine. Local_machine_key is the value set in the Agent Configuration, as described in "Running Vusers Over a Firewall" on page 259 The file will be removed when the LoadRunner Tuning Module agent disconnects from the MI Listener.
- On both UNIX and Windows platforms, the following message will appear in the LoadRunner Tuning Module agent log file: Notify Connected to MI Listener.

Note: The LoadRunner Tuning Module agent tries to connect to the MI Listener machine every Timeout seconds (as defined in the Agent Configuration). After a successful connection, if no Console has connected through this MI Listener to the agent after another Timeout, the LoadRunner Tuning Module will disconnect from the Console.
On a Windows machine, the agent icon's light in the system tray will turn from green to red. On UNIX machines, the file <Local_machine_key>_connected_to_MI_Listener will be removed from the temporary directory in the LoadRunner Tuning Module agent machine.
In both Windows and UNIX, the message Disconnected from MI Listener will appear in the LoadRunner Tuning Module agent log file.

UNIX Connection Errors

After installing the *m_agent_daemon* as described in Chapter 23, "Running Vusers Over a Firewall," you should receive a message: m_agent_daemon installed successfully.

Agent Daemon Errors

ERROR: File m_agent_daemon doesn't exist.

This error means that you are not in the same directory as the file (meaning not in *<LoadRunner_Tuning_Module_root>/bin* directory, or the file really doesn't exist, which indicates a problem with the installation).

WARNING: Could not install m_agent_daemon, reason - user <user_name> is already running m_agent_daemon on this machine.

This warning message occurs when a daemon of this name is already being run by the same user.

ERROR: Could not install m_agent_daemon. Check log file m_agent_daemon[xxx].log in your temp directory.

This error indicates that some error has occurred when loading the daemon. You should check the log file and consult the following troubleshooting tips.

LoadRunner Tuning Module Agent Log File Errors

Error - 10344 : Communication Error: -59961 : Failed to bind a socket while calling bind function.

Error -10344 : Communication Error: -59927 : Failed to create a TCP server for the HTTP channel's server.

Warning -29974 : Failed to create "router" server.

These messages appear because the LoadRunner Tuning Module agent always tries to open port number 443 (because any agent can be a MI Listener, and the MI Listener always listens to this port), and in UNIX machines, this port cannot be opened by any user except for the root user. However, this will not interfere with using this agent for the Load Generator machine.

Error -10343 : Communication error : -59981 : Failed to connect to remote host - <MI_Listener_name> .

The MI Listener is not being run at the time of the connection attempt on the machine set in MI Listener Name in the Agent Configuration.

Error -10343 : Communication error: -59928 : Unresolved server name .

The name passed in MI Listener Name in the Agent Configuration is not a name, full name or IP address of a valid machine, or no value was set.

Error -10343 : Communication error: -59928 : Unresolved server name .

The name passed in Proxy Name in the Agent Configuration is not a name, full name or IP address of a valid machine.

Error -10343 : Communication error: -59945 : Client failed to connect to a PROXY Server with the following settings: (-server_port=<proxy_server_port>)(-server_fd_primary=2)(-server_type=8)(-allowed_msg_size=0)(-allowed_msgs_num=0)(-proxy_configuration_on)(-tcp_tunnel_configuration_on).

The Proxy Name field is empty.

Error -10343 : Communication error: -59982 : Failed to connect to remote host - <MI_Listener_Name>. The remote address is not a valid address.

Error -10343 : Communication error: -59945 : Client failed to connect to a PROXY Server with the following settings: (-server_name=<proxy_server_name>)(-server_port=<proxy_server_port>)(-server_fd_primary=2)(-server_type=8)(-allowed_msg_size=0)(-allowed_msgs_num=0)(-proxy_configuration_on)(-tcp_tunnel_configuration_on).

The Proxy Port set in Agent Configuration, has been set to the wrong port number.

Error -10343 : Communication error: -59913 : NTLM authentication to proxy server error - connection to proxy refused.

The proxy server is configured in for NTLM authentication and the Proxy User Name, Proxy Password and/or Proxy Domain are not set correctly in the Agent Configuration.

Error -10343 : Communication error: - 59880 : Basic authentication to proxy server error - connection to proxy refused.

The proxy server is configured in for Basic authentication and the Proxy User Name and/or Proxy Password are not set correctly in the Agent Configuration.

Error -10343 : Communication error: -59907 : SSL connect error : verify host failed : wrong DNS test .

This error occurs when you have set the Check Server Certificates setting to True, and have not issued a new certificate to the MI Listener machine (see Appendix G, "Working with Digital Certificates" for more details).

Error -10343 : Communication error: -59907 : SSL connect error : certificate verify failed.

Error -10343 : Communication error: -59907 : SSL connect error : sslv3 alert handshake failure.

Error -10343 : Communication error: -59907 : SSL connect error : sslv3 alert bad certificate.

Error -10343 : Communication error: -59907 : SSL connect error : sslv3 alert certificate expired.

These errors occur when you set the Check Server Certificates setting to True. See Appendix G, “Working with Digital Certificates” to learn how to issue a valid certificate.

Error -10343 : Communication error: -59910 : SSL initialization error : Certificate not found .

Error -10343 : Communication error: -59910 : SSL initialization error : No such file or directory.

Error -10343 : Communication error: -59910 : SSL initialization error : system lib.

These errors occur when the Client Certificate owner setting in the Agent Configuration is set to True, but no certificate was installed in the LoadRunner Tuning Module agent machine (see Appendix G, “Working with Digital Certificates” for more details).

MI Listener Log File Errors

Error - 10344 : Communication Error: -59961 : Failed to bind a socket while calling bind function.

Error -10344 : Communication Error: -59927 : Failed to create a TCP server for the HTTP channel’s server.

Warning -29974 : Failed to create "router" server.

This error means that another process on the MI Listener machine is occupying port 443 (for instance the IIS service).

Error -10343 : Communication error: -59904 : SSL accept error : sslv3 alert certificate expired.

These errors occur when you have set the Check Server Certificates setting to True, and the MI Listener's certificate is expired.

Error -10343 : Communication error: -59904 : SSL accept error : sslv3 alert bad certificate.

These errors occur when you have set the Check Server Certificates setting to True, and either:

➤ The MI Listener's certificate does not have a signature that is included in the LoadRunner Tuning Module agent's CA List.

➤ The MI Listener's certificate has a future verification date.

See Appendix G, "Working with Digital Certificates" to learn how to issue a valid certificate and how to add a Certification Authority to a CA list, or how to create a certificate with a new validation date.

Error -10343 : Communication error: -59904 : SSL accept error : peer did not return a certificate.

These errors indicate that the Check Client Certificates setting in the MI Listener Configuration is set to True, but the Client Certificate owner setting in the Agent Configuration is set to False.

Error -10343 : Communication error: -59904 : SSL accept error : no certificate returned.

These errors indicate that the Check Client Certificates setting in the MI Listener Configuration is set to True, and the Client Certificate owner setting in the Agent Configuration is set to True, but either:

➤ The LoadRunner Tuning Module agent's certificate does not have a signature that is included in the MI Listener's CA List.

➤ The LoadRunner Tuning Module agent's certificate has a future verification date.

See Appendix G, "Working with Digital Certificates" to learn how to issue a valid certificate and how to add a Certification Authority to a CA list, or how to create a certificate with a new validation date.

Error -10343 : Communication error: -59904 : SSL accept error : no certificate returned.

These errors indicate that the Check Client Certificates setting in the MI Listener Configuration is set to True, and the Client Certificate Owner setting in the Agent Configuration is set to True, but the LoadRunner Tuning Module agent's certificate has expired.

General Connection Errors

These errors can occur when using all configurations.

If no errors appear both in the LoadRunner Tuning Module agent log, and the MI Listener log, but the agent does not connect to the MI Listener, make sure that the FireWallServiceActive attribute in the Firewall section in the *<LoadRunner_Tuning_Module_Installation>\dat\br_lnch_server.cfg* file on the LoadRunner Tuning Module agent machine, is set to 1.

Verifying Connection Between the Console and Agent through the MI Listener

When there is a successful connection between the LoadRunner Tuning Module agent and the MI Listener, and the Console machine fails to connect, you should check the following:

- The **Name** field in the Load Generators dialog in the Console should match the name set in the **Local Machine Key** in the Agent Configuration.
- The **MI Listener** field in the Load Generators > Details > Firewall tab of the above host matches the name set in the **MI Listener Name** in the Agent Configuration.
- In the Tools menu of the Console, in the Options > Timeout tab, the **Load Generator Connect timeout** might need to be increased, because the Firewalls may slow down the communication.
- Make sure that the Console machine recognizes the LoadRunner Tuning Module agent machine (e.g., by using the ping utility). If this fails, there is a configuration problem in the system not related to LoadRunner Tuning Module, and it must be solved before the connection can be made.
- Make sure that the Console has successfully connected to the MI Listener by checking port 50500 on the MI Listener machine (you can use the netstat utility, on the MI Listener machine).

Troubleshooting Remote Tuning

This section covers some issues you need to handle when using LoadRunner Tuning Module's remote tuning functions.

Not Viewing Information about a Host Running Windows NT

If an NT host machine runs an old version of atl.dll, LoadRunner Tuning Module displays only the host's name without any information about its services. In addition, the following error message is displayed on the host machine:
"The ordinal 57 could not be located in the dynamic link library ATL.DLL."

Solution: If the host machine runs Windows NT, ensure that its version of atl.dll is 3.00.8449 or higher. (The atl.dll file is located in the \\<WINNT installation path>\system32\ directory.)

If the file on the host machine is an old one, do the following:

1. Replace it with a newer version. You can find the new version in the Patches\ATL_patch directory on the CD. Copy it to the c:\winnt\system32 directory on the host machine.
2. Register the new file. (For example, if the atl.dll file is located in C:\WINNT\system32\, register it by running regsvr32 C:\WINNT\system32\atl.dll from the command line.)

Not Viewing Websphere Server Information

To enable you to view (and tune) a WebSphere host from the Console machine, ensure that the WebSphere Administration Server is running on the host machine.

Problems Starting Tuning Agents

If you have trouble starting a tuning agent remotely from the Console, do the following to troubleshoot the problem:

For a Windows Host:

1. On the Console machine, open a DOS window.

2 Execute the ping command against the remote server and see whether you receive a response.

3 Share a network drive or use the net use command to test whether you can mount the remote server.

4 Make sure the Windows Management Instrumentation service is running on the remote server. If you have changed any environment variables (for example, the PATH variable), restart the service.

5 Make sure the java -version command can be executed from the command line on the remote server. Make sure that the Java version is 1.3.1 or later.

For a UNIX Host:

1 On the Console machine, open a DOS window.

2 Execute the ping command against the remote server and see whether you receive a response.

3 On the UNIX server, execute the following command:

```
rsh -l <username> -n "java -version"
```

4 Make sure the rsh daemon is running on the server.

5 On the server, make sure the .rhosts file is updated to allow access from a Windows machine.

6 On the server, make sure the UNIX account's default PATH is configured for launching Java.

7 On the server, execute the following command:

```
rcp.exe "<localfile>" <host>.<username>:<remotefile>
```

Note: Use the appropriate similar commands for ssh & scp (SSL).

C

Working in Expert Mode

Advanced users can fine-tune the LoadRunner Tuning Module configuration settings while working in *Expert Mode*. In Expert mode, additional options are displayed in the Options dialog box and in the Load Generator Information dialog box. This appendix describes the additional settings that are available in the Expert mode:

- Entering Expert Mode
- Options - General Settings
- Options - Debug Information Settings
- Options - Output Settings
- Options - Monitor Settings
- Load Generator Information - UNIX Environment Settings
- Load Generator Information - Connection Log Settings

Entering Expert Mode

The LoadRunner Tuning Module Console Expert mode is intended for support personnel to provide access to system information. When you work in the Expert mode, the Console dialog boxes contain additional options for fine tuning the Console operation.

To activate the Expert mode, choose **Tools** > **Expert Mode.** An active Expert mode is indicated by a check mark.

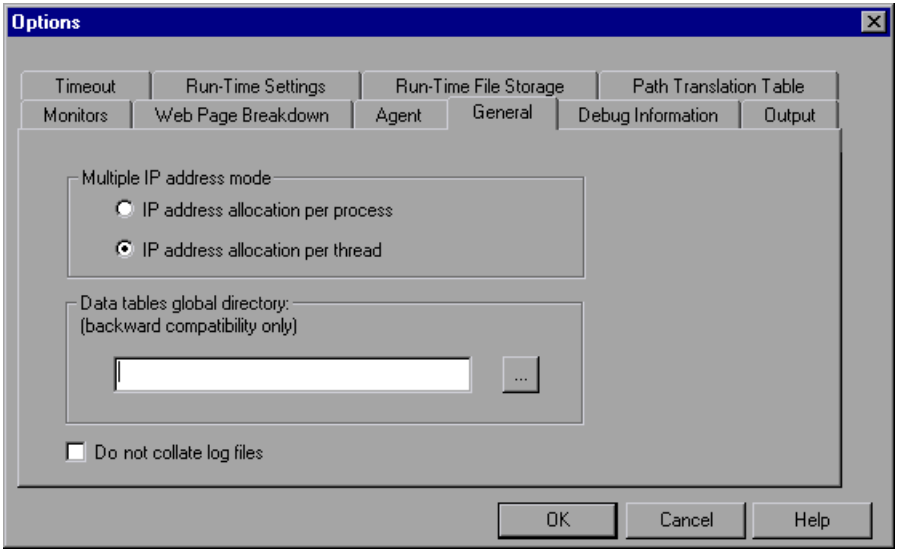
To exit the Expert mode, repeat the above process.

Options - General Settings

The General tab in the Options dialog box allows you to specify global settings for data table storage and multiple IP address allocation, and instruct LoadRunner Tuning Module not to collate log files.

To set the General Expert mode settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **General** tab.



2 Select the Multiple IP address mode.

3 Enter the global directory for data tables.

4 If you want LoadRunner Tuning Module to collate only result files and not log files, check **Do not collate log files**.

5 Click **OK** to accept the settings and close the dialog box.

Understanding the Options - General Tab

The General tab allows you to specify global settings for data table storage, log file collation, and multiple IP address allocation.

Multiple IP address mode: The mode used to allocate IP addresses when the multiple IP address option is enabled (**Session > Enable IP Spoofer**). The Console can allocate an IP address per process or per thread. Web Vusers require IP address allocation per process. WinSock Vuser IP addresses can be allocated per thread or per process. Allocation per thread results in a more varied range of IP addresses in a session.

Data tables global directory: The network location for data tables used as a source for parameter values. This setting is only required for scripts created with earlier versions of LoadRunner Tuning Module.

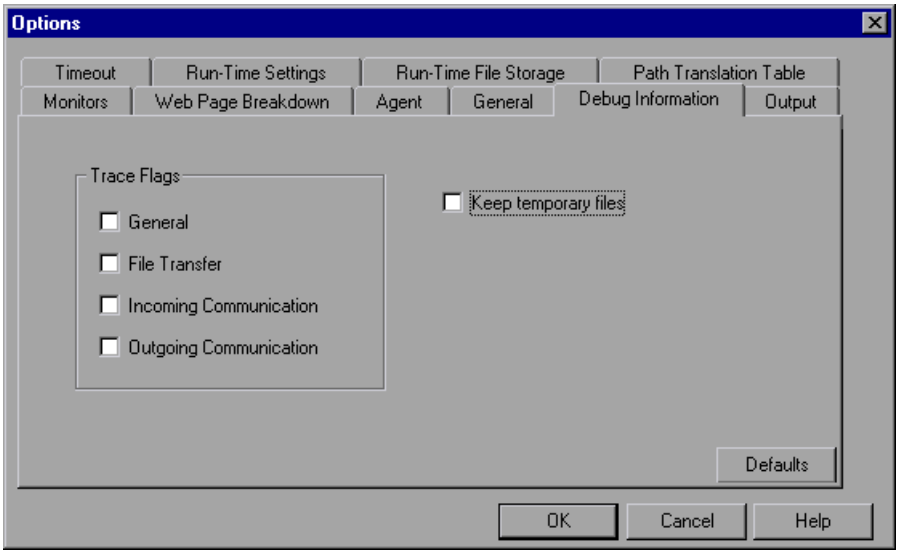
Do not collate log files: Instructs LoadRunner Tuning Module to collate only result files, and not log files.

Options - Debug Information Settings

The Debug settings in the Options dialog box allow you to determine the extent of the trace to be performed during session execution. The debug information is written to the Output window.

To set the Debug Information settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **Debug Information** tab.



2 Select the check boxes for the desired trace flags.

3 To save the temporary run-time files, select the **Keep temporary files** check box.

4 Click **OK** to accept the settings and close the dialog box.

Understanding the Options - Debug Information Tab

Allows you to define the Debug configuration.

Trace flags: For debugging purposes, you can configure the type of trace performed by LoadRunner Tuning Module during test execution. Select the appropriate check box(es) to enable the detailed trace. The trace information appears in the log file located in the specified Agent log directory. The available trace flags are: **General**, **File Transfer**, **Incoming Communication**, and **Outgoing Communication**. You only need to select the flags relating to your problem. For example, if you encounter specific problems with the transfer of files, select the File Transfer flag.

Keep temporary files: The Agent and Console create some temporary files, which collect information such as the parameter file sent to the Vuser, the output compilation file, and the configuration file. The Agent files are saved in **brr** folders in the TMP or TEMP directory of the Agent machine. The Console files are saved in in **lrr** folders in the TMP or TEMP directory of the Console machine. At the end of the session step, all these files are automatically deleted. The **Keep temporary files** setting instructs the Agent and Console not to delete these files if you need them for debugging.

Options - Output Settings

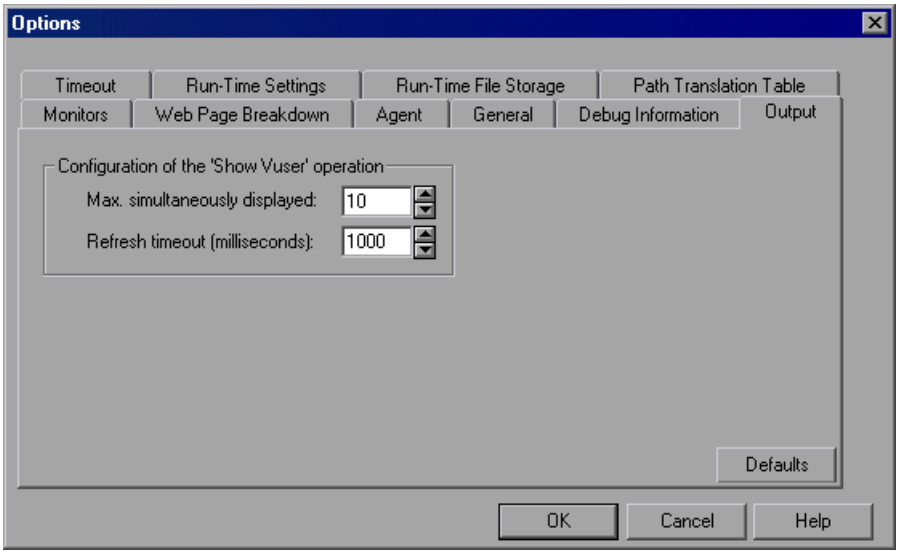
When Expert mode is enabled, the Options dialog box includes the Output tab. This tab contains the following settings:

Max simultaneously displayed: Specifies the maximum number of Vuser logs that may be displayed simultaneously, as well as the maximum number of active UNIX, GUI, RTE, or Web Vusers that the Console should display by opening up Run-Time Viewers on your machine. The default number is 10.

Refresh timeout: Defines how often to refresh the Vuser log. The default is every 1000 milliseconds.

To set the Output settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **Output** tab.



2 Specify the maximum number of Vuser logs to be displayed simultaneously, in the **Max. simultaneously displayed** box.

3 Specify the frequency at which LoadRunner Tuning Module refreshes the Vuser log, in the **Refresh timeout** box.

4 Click **OK** to accept the settings and close the dialog box.

Understanding the Options - Output Tab

The Output tab enables you to configure the display of running Vusers on the Console machine.

Configuration of the 'Show Vuser' Operation:

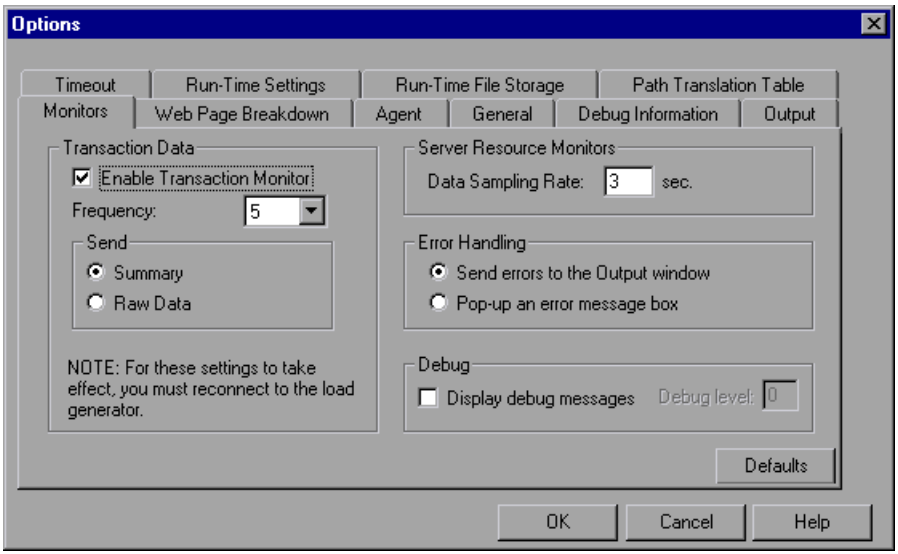
➤ **Max. simultaneously displayed:** Specifies the maximum number of Vuser logs that may be displayed simultaneously, as well as the maximum number of active UNIX, GUI, RTE, or Web Vusers that the Console should display by opening up Run-Time Viewers on your machine. The default number is ten.

➤ **Refresh timeout (milliseconds):** Defines how often to refresh the Vuser log. The default is every 1000 milliseconds.

Options - Monitor Settings

Expert mode provides the following additional monitor setting:

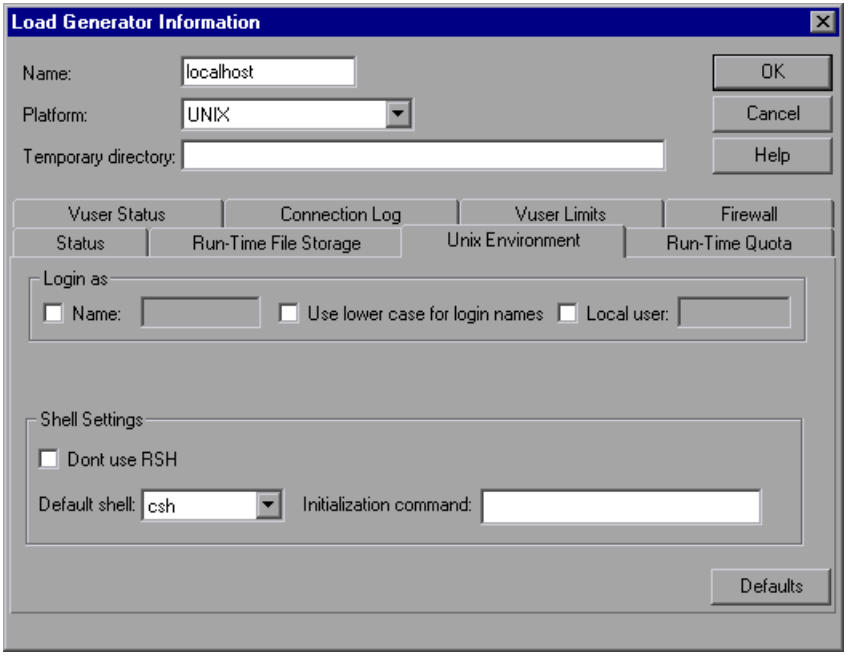
Send Summary or Raw Data: sends a summary of the data collected back to the Console, or sends all of the data in raw form. Sending the data in raw form saves time because the data does not need to be processed. However, since all of the data is being transferred to the Console, it may cause more network traffic. If the transfer speed is significant to you, it is recommended that you choose **Summary**.



Load Generator Information - UNIX Environment Settings

Expert mode provides the following additional UNIX Environment setting:

Local User: UNIX load generators that use the *rsh* shell establish a connection as the current NT user (due to security considerations). To "mislead" rsh and log in as a user other than the current NT login, select the **Local user** check box and specify the desired UNIX login name. Since modifying the local user name is a security breach for *rsh*, this option should only be used when you encounter a problem connecting to the remote machine.



Load Generator Information - Connection Log Settings

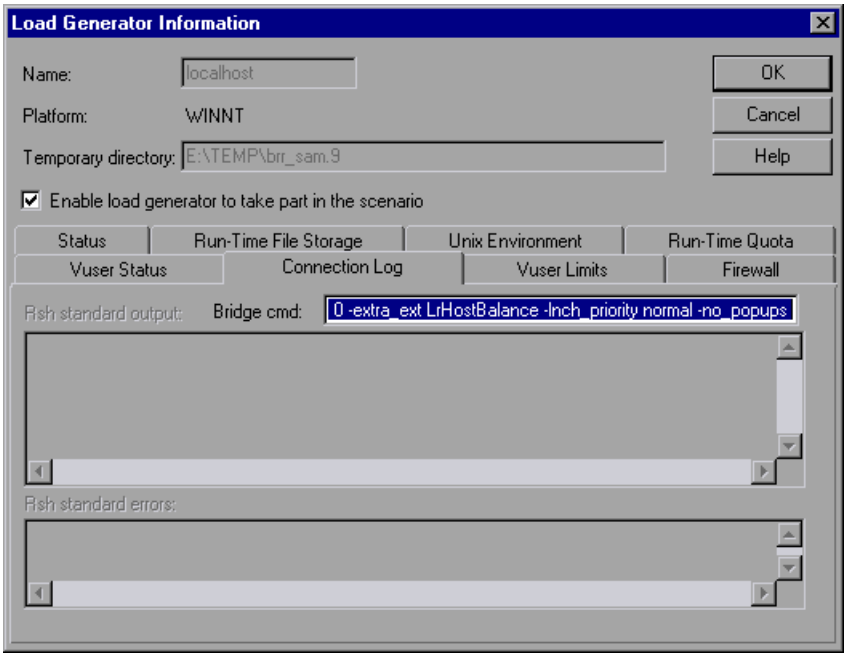
The Connection Log tab in the Load Generator dialog box allows you to view the standard output and standard errors generated as the Console connects to the selected UNIX load generator. You can also change the command that the Console sends to the remote bridge in order to connect to the load generator.

To set the Connection Log settings:

1 Click the **Generators** button, or select **Session** > **Load Generators**. The Load Generators dialog box opens.

2 Click **Connect** to change the Status of a load generator from Down to Ready.

3 Click the **Details** button. The Load Generator Information dialog box opens. Select the **Connection Log** tab.



Rsh standard output: Displays rsh standard output generated as the Console sends the connection command to the selected UNIX load generator.

Bridge cmd: Enter a new command if you want to change the default bridge command being sent by the Console to the remote bridge in order to connect the UNIX load generator

Rsh standard errors: Displays rsh standard errors as the Console connects to the selected UNIX load generator.

D

Performing Path Translation

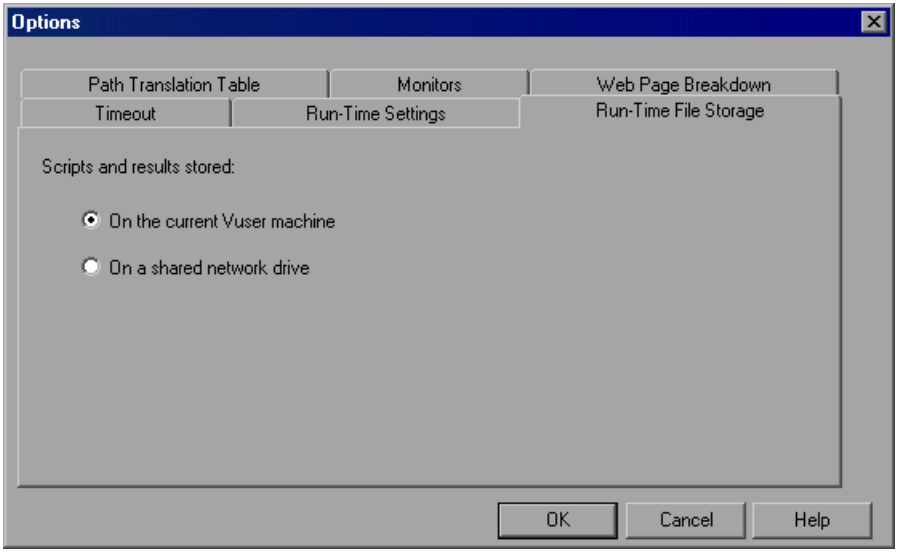
When you run a session step, LoadRunner Tuning Module gathers run-time data from the participating Vusers. By default, LoadRunner Tuning Module stores the data in temporary files on each Vuser machine. After the session step, the data is collated in the general results directory.

Alternatively, you can instruct LoadRunner Tuning Module to write the run-time data directly to a shared network drive. (See Chapter 14, "Configuring Session Steps.") This method is not recommended, since it increases network traffic and necessitates path translation.

Understanding Path Translation

Path Translation is a mechanism used by LoadRunner Tuning Module to convert a remote path name for the Console. A typical session step might have the LoadRunner Tuning Module Console running on a Windows-based machine and include multiple Vusers running on both Windows-based and UNIX load generators. One remote load generator may map the network drive as *F*, while another load generator maps the same drive as *H*. In a complex session step such as this, you need to ensure that all participating machines recognize the same network drive.

You instruct LoadRunner Tuning Module to store scripts and run-time data results on a shared network drive from the Run-time File Storage tab of the Options dialog box.



Result and script files stored on a shared network drive require you to perform path translation.

The Script view contains a list of all the Vuser scripts associated with a session step—and their locations. A script's location (path) is always based on the Console machine's mapping of that location. If a Vuser load generator maps to the script's path using a different name, path translation is required.

For example, assume that the Console is running on a Windows-based machine named *pc2*, and that a Vuser script is located on a network drive. The Console machine maps the network drive as *m*:\lr_tests. If the remote Vuser machine (load generator) hosting the Vusers also maps the path as *m*:\lr_tests, no translation is necessary. However, if the remote machine maps the path as another drive or path, for example *r*:\lr_tests, you must translate the path to enable the load generator to recognize the script location.

Similarly, when saving run-time result files to a shared drive that is mapped differently by the Console and remote load generator, you must perform path translation.

Path translation is also effective across platforms—between Windows and UNIX. You use path translation to translate Windows-based paths (as seen by the Console) into paths recognized by the UNIX Vuser load generator.

Adding Entries to the Path Translation Table

To translate a path from one Windows-based computer to another, or between Windows-based and UNIX machines, you create an entry in the Path Translation table. This table contains a list of paths translated into formats that can be recognized by different machines.

Each line of the Path Translation table has the following format:

<console_host><console_path><remote_path>[<remote_host>]

console_host The name or type of the machine that is running the Console. For example, if the Console is running on a Windows-based computer, you could type win in the host field. Alternatively, you could enter the name of the machine running the Console (for example, LOADPC1).

The value of *console_host* can be:

hostname	the name of the machine running the Console
win	the Console is running on a Windows-based computer
unix	the Console is running on a UNIX machine
all	the Console is running on a Windows-based or a UNIX machine

console_path	The path of a specific directory—as recognized by the Console. For example, if the directory *scripts* is located on the network drive *r*—as mapped by the Console—type the path r:\scripts in the *console_path* field.
remote_path	The path of a specific directory—as recognized by the remote machine. For example, if the directory *scripts* is located on the network drive *n—as mapped by the remote load generator*—type the path n:\scripts in the *remote_path* field. If a Vuser on the remote UNIX load generator recognizes the above path as /m/tests, you would type this path in the *remote_path* field.
remote_host	The name or type of the remote load generator. For example, if all the remote machines are UNIX workstations, you could type unix in the *remote_host* field. The options for the *remote_host* field are the same as the options for the *console_host* field, listed above. The *remote_host* parameter is optional.

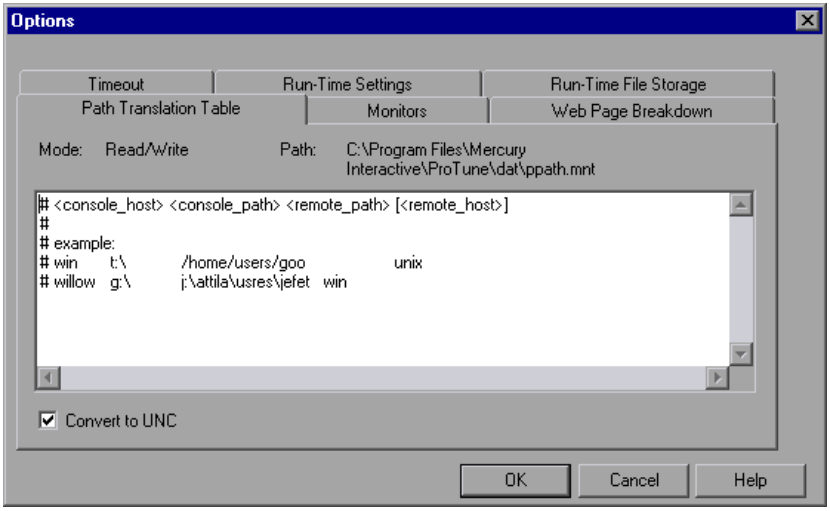
Editing the Path Translation Table

You maintain the Path Translation table using the LoadRunner Tuning Module Console. LoadRunner Tuning Module saves the Path Translation table as an ASCII file, *ppath.mnt*. This file, stored in LoadRunner Tuning Module_*directory/dat*, has a one–line entry for each network path to translate.

To edit the Path Translation table:

1 Start the LoadRunner Tuning Module Console.

2 Choose **Tools** > **Options** and select the **Path Translation Table** tab. The Path Translation Table view opens.



3 Before you enter path translation information, consider using the Universal Naming Convention method. If your machines are Windows machines, you can tell the Console to convert all paths to UNC, and all machines will be able to recognize the path without requiring path translation. An example of UNC format is \\machine_a\results.

Select the Convert to UNC check box to tell LoadRunner Tuning Module to ignore the path translation table and to convert all paths to the Universal Naming Convention.

4 If your machines are not Windows machines and you require path translation, type the path information into the table. You can insert comments by typing the "#" symbol at the start of a line in the table.

5 Click **OK** to close the table and save the information.

Path Translation Examples

The following section illustrates sample Path Translation Table entries.

Note that when you translate a Windows-based path to a UNIX path, you must enter the appropriate slashes—forward slashes for UNIX and back slashes for Windows-based paths.

The examples below show the use of the Path Translation table for a Windows-based Console called Merlin.

In the first example, Vusers are running on a Windows 2000 machine, Oasis. Merlin maps the network drive as f:, while Oasis maps it as g:\loadtest.

merlin	f:\	g:\loadtest\	Oasis

In the second example, Vusers are running on a UNIX machine, Ultra. Ultra maps the networks drive as /u/tests/load.

merlin	f:\	/u/tests/load/	Ultra

In the third example, the mapping of the network drive by the remote load generator Jaguar, is identical to the Console's mapping, so no translation is required. This line can be excluded from the Path Translation table.

merlin	n:\	n:\	Jaguar

In the fourth example, all Windows-based Vuser load generators map the network drive as m:\loadtest.

merlin	l:\mnt\	m:\loadtest\	win

E

Working with Server Monitor Counters

When you configure the System Resource, Microsoft IIS, Microsoft ASP, ColdFusion, and SQL Server monitors, you are presented with a list of default counters that you can measure on the server you are monitoring. Using the procedure described below, you can create a new list of default counters by including additional counters, or deleting existing counters.

In addition, there are specific counters that are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a server.

The following sections describe:

- Changing a Monitor's Default Counters
- Useful Counters for Stress Testing

Changing a Monitor's Default Counters

You can change the default counters for the System Resource, Microsoft IIS, Microsoft ASP, ColdFusion, or SQL Server monitors by editing the *res_mon.dft* file found in the LoadRunner Tuning Module/dat directory.

To change the default counters:

1. Open a new session and click the **Session** tab.
2. For each of the monitors, select the counters you want to measure.
3. Save the session and open the session *.lrs* file with an editor.

4 Copy the MonItemPlus section of the each counter you selected into the *res_mon.dft* file.

5 Count the number of new counters in the file and update the **ListCount** parameter with this number.

Useful Counters for Stress Testing

Certain counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a server.

The following is a list of counters that are useful for monitoring Web server performance:

Object	Counter
Web Service	Maximum Connections
Web Service	Bytes Total/sec
Web Service	Current NonAnonymous Users
Web Service	Current Connections
Web Service	Not Found Errors
Active Server Pages	Requests/sec
Active Server Pages	Errors/sec
Active Server Pages	Requests Rejected
Active Server Pages	Request Not Found
Active Server Pages	Memory Allocated
Active Server Pages	Requests Queued
Active Server Pages	Errors During Script Run Time
Memory	Page Faults/sec
Server	Total Bytes/sec
Process	Private Bytes/Inetinfo

The following is a list of counters that are useful for monitoring SQL Server performance:

Object	Counter
SQLServer	User Connections
SQLServer	Cache Hit Ratio
SQLServer	Net-Network Reads/sec
SQLServer	I/O-Lazy Writes/sec
SQLServer-Locks	Total Blocking Locks
PhysicalDisk	Disk Queue Length

The following is a list of counters that are useful for monitoring both Web and SQL server performance:

Object	Counter
Processor	% Total Processor Time
PhysicalDisk	% Disk Time
Memory	Available Bytes
Memory	Pool Nonpaged Bytes
Memory	Pages/sec
Memory	Committed Bytes
System	Total Interrupts/sec
Object	Threads
Process	Private Bytes:_Total

Note: The % Disk Time counter requires that you run the diskperf -y utility at the command prompt and reboot your machine.

F

Configuring Multiple IP Addresses

When you run a session step, the Vusers on each load generator machine use the machine's IP address. You can define multiple IP addresses on a load generator machine to emulate a real-life situation in which users sit on different machines.

This appendix describes:

- Adding IP Addresses to a Load Generator
- Using the IP Wizard
- Configuring Multiple IP Addresses on UNIX
- Updating the Routing Table
- Enabling Multiple IP Addressing from the Console

About Multiple IP Addresses

Application servers and network devices use IP addresses to identify clients. The application server often caches information about clients coming from the same machine. Network routers try to cache source and destination information to optimize throughput. If many users have the same IP address, both the server and the routers try to optimize. Since Vusers on the same load generator machine have the same IP address, server and router optimizations do not reflect real-life situations.

LoadRunner Tuning Module's multiple IP address feature enables Vusers running on a single machine to be identified by many IP addresses. The server and router recognize the Vusers as coming from different machines and as a result, the testing environment is more realistic.

Note: The maximum number of IP addresses that can be spoofed per network card for Windows NT SP3 is 35 IPs; Solaris (version 2.5.1) up to 255 IPs; Solaris (version 2.6 and higher) up to 8192 IPs.

Applicable Protocols

The multiple IP address feature is applicable to the following protocols:

- **Client/Server:** DNS, Windows Sockets
- **Custom:** Java Vuser, Javascript Vuser, VB Vuser, VB Script Vuser
- **E-business:** FTP, Palm, SOAP, Web (HTTP/HTML) protocols, WinSock\Web Dual Protocol
- **ERP:** Oracle NCA, PeopleSoft 8 multi-lingual, Siebel-Web
- **Mailing Services:** Internet Messaging (IMAP), MS Exchange (MAPI), POP3, SMTP
- **Streaming Data:** Real
- **Wireless:** i-Mode, VoiceXML, WAP

This feature can be implemented on Windows and UNIX platforms.

Adding IP Addresses to a Load Generator

LoadRunner Tuning Module includes an IP Wizard program that you run on each Windows NT or Windows 2000 load generator machine to create multiple IP addresses. You add new IP addresses to a machine once and use the addresses for all session steps. For information about adding IP addresses on UNIX machines, see "Configuring Multiple IP Addresses on UNIX" on page 786.

The following procedure summarizes how to add new IP addresses to a load generator:

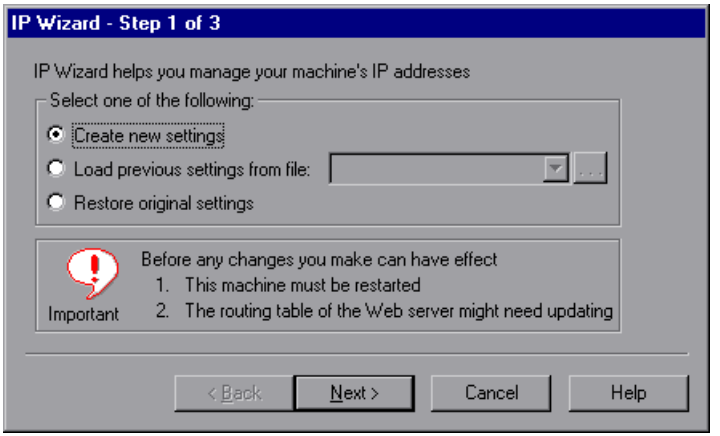
1. Run the IP Wizard on the load generator machine to add a specified number of IP addresses. Manually configure the new IP addresses for UNIX load generator machines.
2. Restart the machine.
3. Update the server's routing table with the new addresses, if necessary.
4. Enable this feature from the Console. Refer to "Enabling Multiple IP Addressing from the Console" on page 788.

Using the IP Wizard

The IP Wizard resides on each load generator machine. You run this process once to create and save new IP addresses on Windows machines. The new addresses can be a range of addresses defined by the Internet Assignment Numbers Authority. They are for internal use only, and cannot connect to the Internet. This range of addresses is the default used by the IP Wizard.

To add new IP addresses to a load generator machine:

1 Invoke the IP Wizard by clicking **Start** > **Programs** > LoadRunner Tuning Module > **Tools** > **IP Wizard**.

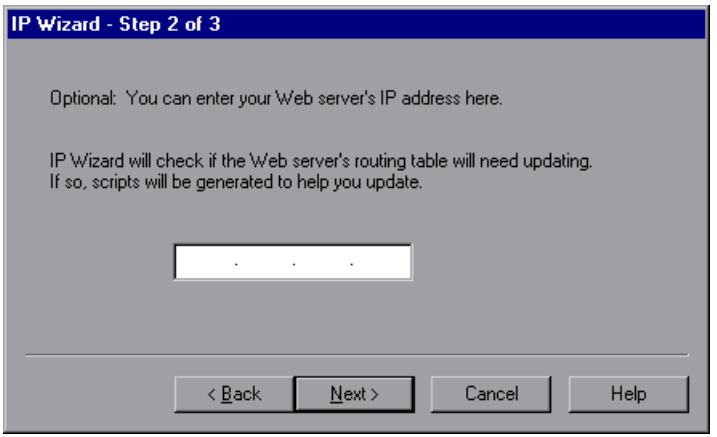


2 If you have an existing file with IP address settings, select **Load previous settings from file** and choose the file.

3 If you are defining new settings, select **Create new settings**.

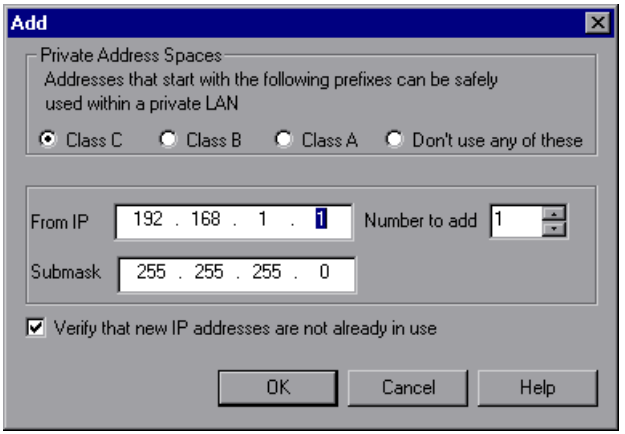
4 Click **Next** to proceed to the next step. If you have more than one network card, choose the card to use for IP addresses and click **Next**.

The optional Web server IP address step enables the IP Wizard to check the server's routing table to see if it requires updating after the new IP addresses are added to the load generator.



5 To check the server's routing table directly after adding the addresses, enter the server IP address. Refer to "Updating the Routing Table" on page 787 for more information.

6 Click **Next** to see a list of the machine's IP address(es). Click **Add** to define the range of addresses.



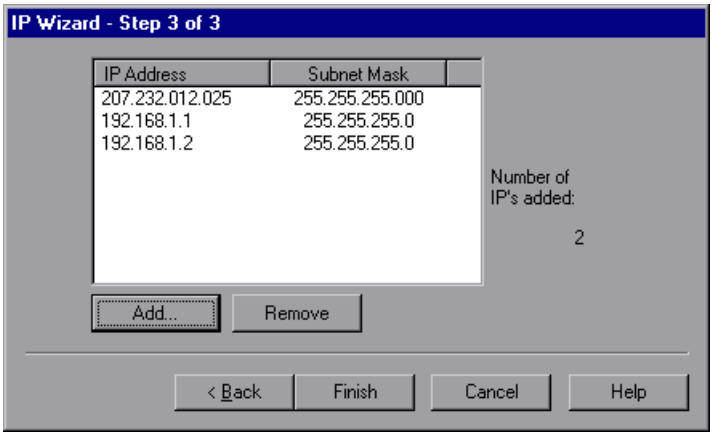
IP addresses include two components, a *netid* and *hostid*. The submask determines where the netid portion of the address stops and where the hostid begins.

7 Select a class that represents the correct submask for the machine's IP addresses.

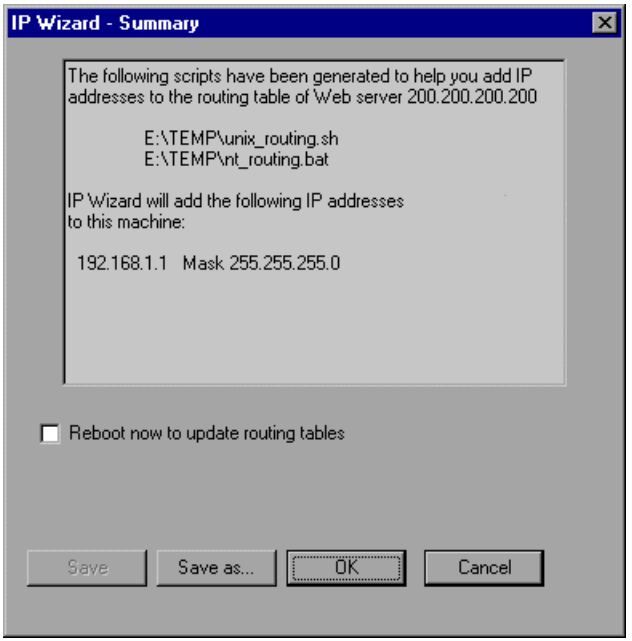
8 Specify the number of addresses to create. Select **Verify that new IP addresses are not already in use** to instruct the IP Wizard to check the new addresses. The IP Wizard will only add the addresses not in use.

9 Click **OK** to proceed.

After the IP Wizard creates the new addresses, the summary dialog box lists all of the IP addresses.



10 Click **Finish** to exit the IP Wizard. The IP Wizard Summary dialog box is displayed.



11 Note the address of the *.bat* file, and see "Updating the Routing Table" on page 787 for information about using the batch file to update the routing table, if necessary.

12 After you update the routing table, check **Reboot now to update routing tables** to initialize the NT device drivers with the new addresses.

13 Click **OK**.

Configuring Multiple IP Addresses on UNIX

To configure multiple IP addresses on UNIX, manually configure the addresses on the load generator machine.

Solaris 2.5, 2.6, 7.0, 8.0

To configure the hme0 device to support more than one IP address:

1 Create entries in /etc/hosts for each hostname on your physical machine:

```
128.195.10.31  myhost
128.195.10.46  myhost2
128.195.10.78  myhost3
```

2 Create */etc/hostname.hme0:n* files that contain the hostname for the virtual host *n*. Note that *hostname.hme0:0* is the same as *hostname.hme0*.

```
/etc/hostname.hme0   (Contains name myhost)
/etc/hostname.hme0:1 (Contains name myhost2)
/etc/hostname.hme0:2 (Contains name myhost3)
```

The above changes will cause the virtual hosts to be configured at boot time.

3 You can also directly enable/modify a logical hosts configuration by running *ifconfig* directly on one of the logical hosts, using the *hme0:n* naming scheme:

```
% ifconfig hme0:1 up
% ifconfig hme0:1 129.153.76.72
% ifconfig hme0:1 down
```

To verify the current configuration, use *ifconfig –a*.

Linux

To define multiple IP addresses for a single Ethernet card, you need IP Aliasing compiled into the kernel. To do this, use the *ifconfig* command:

```
/sbin/ifconfig eth0:0 x.x.x.x netmask 255.255.x.x up
```

Substitute the new IP address for x.x.x.x, and insert the correct information for subnet mask. Place this command in the *rc.local* file so that it executes upon boot.

HP 11.0 or higher

To define multiple IP addresses for a single Ethernet card, you need IP Aliasing compiled into the kernel. To do this, use the *ifconfig* command:

```
/sbin/ifconfig lan1:0 x.x.x.x netmask 255.255.x.x up
```

Substitute the new IP address for x.x.x.x, and insert the correct information for subnet mask. Place this command in the *rc.local* file so that it executes upon boot.

Updating the Routing Table

Once the client machine has new IP addresses, the server needs the addresses in its routing table, so that it can recognize the route back to the client. If the server and client share the same netmask, IP class, and network, the server's routing table does not require modification.

Note: If there is a router between the client and server machines, the server needs to recognize the path via the router. Make sure to add the following to the server routing table: route from the Web server to the router, and routes from the router to all of the IP addresses on the load generator machine.

To update the Web server routing table:

1 Edit the batch file that appears in the IP Wizard Summary screen. An example *.bat* file is shown below.

```
REM This is a bat file to add IP addresses to the routing table of a
server
REM Replace [CLIENT_IP] with the IP of this machine that the server
already recognizes
REM This script should be executed on the server machine

route ADD 192.168.1.50 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.1.51 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.1.52 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.1.53 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.1.54 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
```

2 For each occurrence of **[CLIENT_IP]**, insert your IP address instead.

3 Run the batch file on the server machine.

Enabling Multiple IP Addressing from the Console

Once you define multiple IP addresses, you set an option to tell the Console to use this feature.

To enable multiple IP addressing from the Console:

1 In the Console Design view, select **Session** > **Enable IP Spoofer**.

Note: You must select this option before connecting to a load generator.

---

2 Use the **General Options** of the Console Expert Mode to specify how the Console should implement this feature.

For more information, refer to Appendix C, "Working in Expert Mode."

G

Working with Digital Certificates

A Digital Certificate is an electronic "credit card" that establishes your credentials when doing business or other transactions on the Web. It is issued by a Certification Authority (CA). It contains the IP address of the machine for which it was issued, a validation date, and the digital signature of the certificate-issuing authority.

This appendix describes:

- Using Digital Certificates with Firewalls
- Creating and Using Digital Certificates

Using Digital Certificates with Firewalls

When the MI Listener sends its Public Key to the LoadRunner Tuning Module agent, it always sends its certificate as well (this is the server-side certificate). The LoadRunner Tuning Module agent can be configured to authenticate the certificate which it received, as described in Chapter 23, "Running Vusers Over a Firewall." If the agent is configured to authenticate the certificate, it can verify whether the sender is really the machine that it claims to be by:

- Comparing the certificate's IP address with the sender's IP address.
- Checking the validation date.
- Looking for the digital signature in its Certification Authorities list.

The MI Listener may also require the LoadRunner Tuning Module agent to send a certificate at any point in the session. This is called the client-side certificate, as described in the MI Listener Configuration Settings in Chapter 23, "Running Vusers Over a Firewall." If the LoadRunner Tuning Module agent owns a certificate, it sends it to the MI Listener for the same authentication process. If the LoadRunner Tuning Module agent does not own a certificate, the communication might not be continued.

An SSL CA list and an SSL Certificate are included in each LoadRunner Tuning Module installation. This certificate is the same for all LoadRunner Tuning Module installations, which means that it can be obtained by third parties. Therefore, if you are interested in a more secure process, you should create your own Certificate Authority and include it in the list, and issue matching certificates for your machines.

Creating and Using Digital Certificates

You create a Certification Authority using the gen_ca_cert.exe (on UNIX platforms gen_ca_cert) utility, and a Digital Certificate using the gen_cert.exe (on UNIX platforms gen_cert) utility. Both utilities can be used on UNIX and Windows platforms, using a command-line interface.

To creating a Certificate Authority using gen_ca_cert:

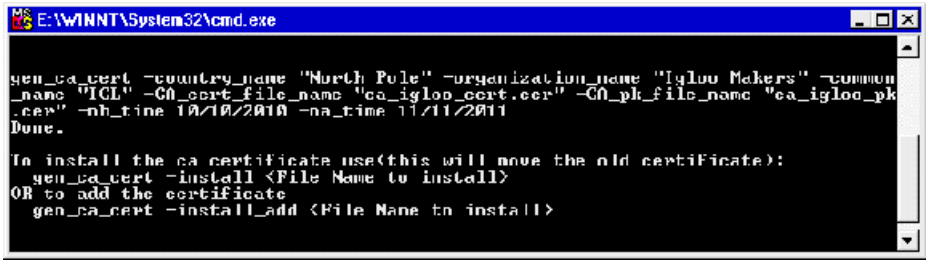
1 To view the format and usage, run the *gen_ca_cert* utility from the <LoadRunner Tuning Module root folder>\launch_service\bin directory.

2 Create a new Certificate Authority by running the gen_ca_cert command with at least one of the options: -country_name <country name> -organization_name <organization name> and -common_name <the name of the CA>.

This process creates two files in the directory from which the utility was run: the CA Certificate (cacert.cer), and the CA Private Key (capvk.cer). To provide different file names, use the -CA_cert_file_name and the -CA_pk_file_name options respectively.

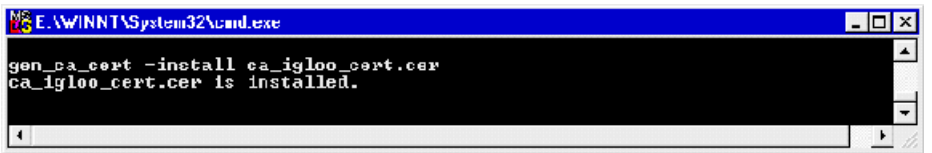
By default, the CA is valid for three years, from the time that the CA is generated. To change the validation dates, use the options -nb_time <beginning of validity in dd/mm/yyyy format> and/or -na_time <ending of validity in dd/mm/yyyy format>.

The following example creates two files: *ca_igloo_cert.cer* and *ca_igloo_pk.cer* in the current directory.:



3 To install this CA, use the -install <name of certificate file> option. This option replaces any previous CA list and creates a new one that includes only this CA.

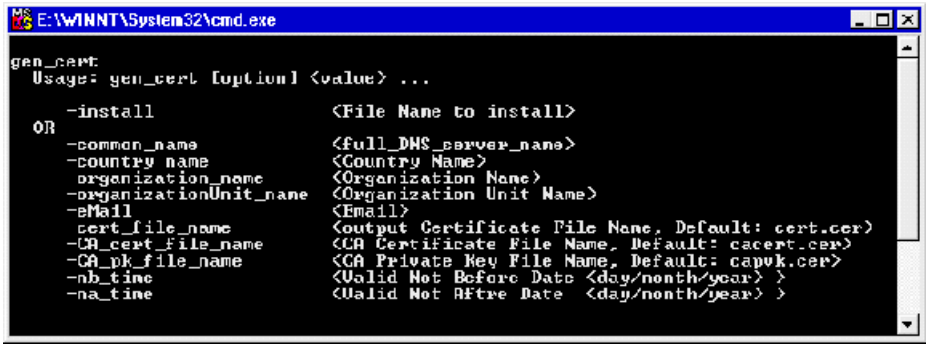
To add the new CA to the existing CA list, use the -install_add <name of certificate file>.



4 The -install and -install_add options install the certificate file only. Keep the private key file in a safe place and use it only for issuing certificates.

To create a Digital Certificate using gen_cert:

1 To view the format and usage, run the *gen_cert* utility from the <LoadRunner Tuning Module *root folder*>*launch_service**bin* directory.



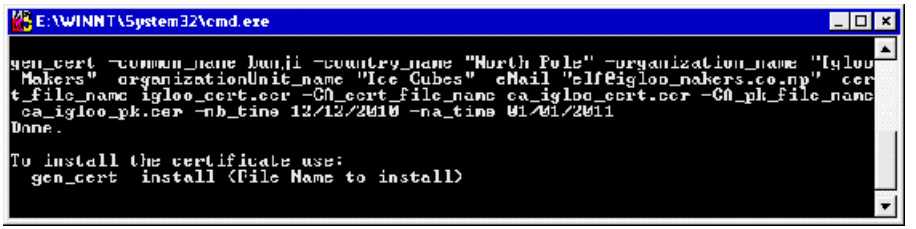
2 Create a new Digital Certificate by running the gen_cert command with at least one of the options: -country_name <country name>, -organization_name <organization name>, -organization_unit_name <organization unit name>, -eMail <email address> and -common_name <the name, full name or IP address of the machine>.

The CA Certificate and the CA Private Key files are necessary for the creation of the certificate. By default, it is assumed that they are in the current directory, and are named *cacert.cer* and *capvk.cer* respectively. In any other case, use the -CA_cert_file_name and -CA_pk_file_name options to give the correct files and locations.

In this process, the certificate file is created in the directory from which the utility was run. By default, the file name is *cert.cer*. To provide a different name, use the -cert_file_name option.

By default, the CA is valid for three years, from the time that the CA is generated. To change the validation dates, use the -nb_time <beginning of validity in dd/mm/yyyy format> and/or -na_time <ending of validity in dd/mm/yyyy format> options .

The following example creates the *igloo_cert.cer* file in the current directory:



3 If you wish to install this certificate, use the -install <name of certificate file> option. This option replaces any previous certificate, as it is possible to own only one certificate per machine.

Host Resolution Functions Copyright Agreement

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

S

T

U

V

W

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089
Tel. (408)822-5200 (800) TEST-911
Fax. (408)822-5300

Main Telephone: (408) 822-5200
Sales & Information: (800) TEST-911, (866) TOPAZ-4U
Customer Support: (877) TEST-HLP
Fax: (408) 822-5300

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