Mercury Tuning Console

USER'S GUIDE

Mercury Tuning Console

User's Guide

Version 8.0

MERCURY INTERACTIVE

Mercury Tuning Console User’s Guide, Version 8.0

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

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

TMCONUG8.0/01

Table of Contents

PART III: EXECUTING A TUNING SESSION

PART IV: WORKING WITH FIREWALLS

PART V: WORKING WITH DIAGNOSTICS

PART VI: MONITORING A SESSION

PART VII: REMOTE TUNING

PART VIII: APPENDIXES

Welcome to Mercury Tuning Module

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

Mercury 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. Mercury 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, Mercury Tuning Module helps the more advanced user to explore and tune the business processes by providing a simple and organized methodology.

Online Resources

Mercury Tuning Module includes the following online tools:

Read Me First provides last-minute news and information about Mercury 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 Mercury Tuning Module online books.

Mercury Tuning Module **Function Reference** gives you online access to all of Mercury 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*.

Mercury Tuning Module **Context Sensitive Help** provides immediate answers to questions that arise as you work with Mercury Tuning Module. It describes dialog boxes, and shows you how to perform Mercury 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 Mercury 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.

Mercury Tuning Module Documentation Set

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

- install Mercury Tuning Module
- use the Mercury Tuning Module Console

Using the Mercury Tuning Module Documentation Set

The Mercury Tuning Module documentation set consists of an installation guide, a Console user's guide, and a Monitor Reference.

In addition to the Mercury Tuning Module documentation set, refer to the *Mercury Virtual User Generator User's 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 Mercury Tuning Module, refer to the *LoadRunner Installation Guide*. The installation guide explains how to install:

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

Console User's Guide

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

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

Monitor Reference

The Mercury Tuning Module documentation pack includes one Monitor Reference guide:

The *LoadRunner Monitor Reference* describes how to set up the server monitor environment and configure LoadRunner monitors for monitoring data generated during a tuning session.

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.

Guide for Creating Vuser Scripts

The LoadRunner documentation pack includes one guide for creating scripts:

The *Mercury Virtual User Generator User's Guide* describes how to create scripts using VuGen. When necessary, supplement this document with the online *LoadRunner Function Reference* and the *WinRunner User's Guide* for creating GUI scripts.

For information on	Look here...
Installing Mercury Tuning Module	*LoadRunner Installation Guide*
The Mercury Tuning Module testing process	*Mercury Tuning Module Console User's Guide*
Creating Vuser scripts	*Mercury Virtual User Generator User's Guide*
Creating and running sessions	*Mercury Tuning Module Console User's Guide*
Configuring the server monitors	*LoadRunner Monitor Reference*
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 Mercury 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 documentation for the Virtual User Generator, Function Reference, and Analysis contains several LoadRunner-specific terms. The following table lists the LoadRunner terms and their parallel ones in the Mercury Tuning Module.

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

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

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

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 Mercury Tuning Module

1

Introducing Mercury Tuning Module

Mercury 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 Mercury 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:** Mercury 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:** Mercury Tuning Module pinpoints the precise areas where bottlenecks may occur, by using component tests custom-designed by experts.
- **Expert Recommendations:** Mercury 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:** Mercury 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 chapter describes:

- Mercury Tuning Module’s Features
- LoadRunner License Information

Mercury Tuning Module’s Features

Mercury 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

Mercury 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 29.

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 Mercury Tuning Module to automatically select the monitors most critical for each component's performance. See "Automatically Assigning Monitors and Alerts" on page 40.

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, Mercury Tuning Module can halt or scale down the test, or stop increasing the load on the component.

In addition, Mercury 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, Mercury Tuning Module locates and displays the cause of the problem, giving you one clear, comprehensive picture. See "Defining Alerts" on page 149.

Canned Scripts

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

Mercury 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 Mercury Tuning Module's recommended scripts, you can also create custom scripts for your application.

See "Using Built-In Knowledge" on page 89.

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 244.

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 245.

Recommended Values Scanning

Mercury Tuning Module compares the current values on a service or server with the recommended values in Mercury Tuning Module's knowledge base, and shows you those values that differ from the recommended ones. See "Scan Recommended Values Step" on page 253.

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 Mercury Tuning Module, so you can restore your system to its original state. See "Introducing Remote Tuning" on page 383.

Cartridges

Mercury 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 89.

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 239.

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 414.

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 Mercury Tuning Module's Help menu.

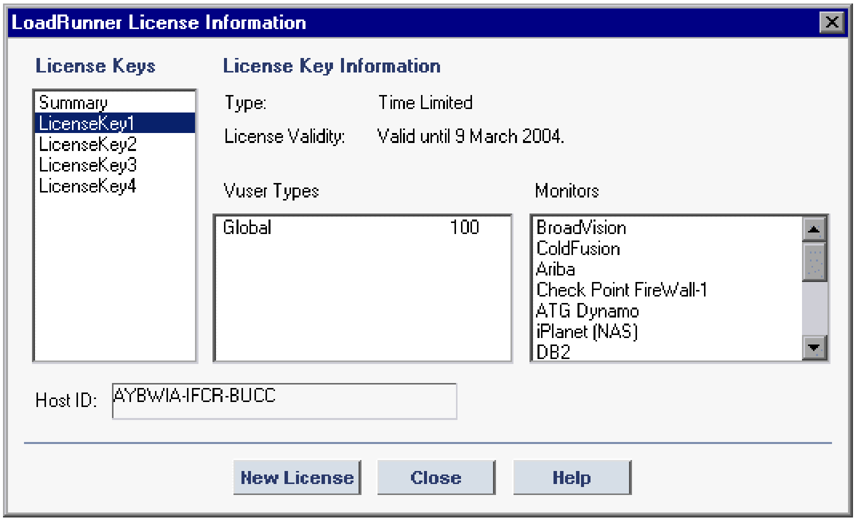
Flexibility and Scalability

To provide maximum flexibility and scalability, Mercury 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.

LoadRunner License Information

To preview your license key information, in the Mercury LoadRunner window (**Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner**), click **License**.

The License Information dialog opens.



LoadRunner License Information

License Keys: Summary, LicenseKey1, LicenseKey2, LicenseKey3, LicenseKey4

License Key Information

Type: Time Limited

License Validity: Valid until 9 March 2004.

Vuser Types	
Global	100

Monitors: BroadVision, ColdFusion, Ariba, Check Point FireWall-1, ATG Dynamo, iPlanet (NAS), DB2

Host ID: AYBWIA-IFCR-BUCC

The License Information dialog displays the following information:

License Information

License Keys: Displays the available license keys, as well as a summary of all the available license keys.

License Key Information

Type: Displays the type of license available for the license key you selected. The following types of licenses are available:

- **Permanent:** The license never expires.
- **Time Limited:** The license is limited by a start date and an expiration date.

- **Temporary:** The license is granted for a pre-defined number of days after product installation.
- **Plugged:** The license requires a dongle.
- **VUD-based:** The license is limited by a number of Virtual User Days (VUDs). A VUD license enables the user to use the product an unlimited number of times within a period of 24 hours.

Note: If a user has a VUD-based license for 1000 Vusers and the maximum number of concurrently running Vusers within a 24-hour period is 300, the following day the user will be able to run the remaining 700 Vusers.

License Validity: Displays the time limitation of the selected license key.

Vuser Types: Displays a list of Vuser protocols available for the selected license key.

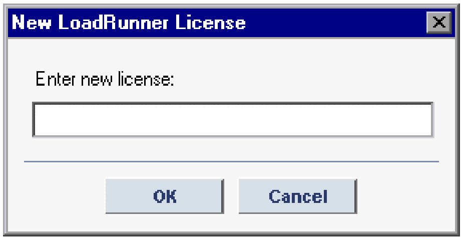
Monitors: Displays the server monitors available for the selected license key.

Host ID: Displays an ID for a specific machine. To receive a license key for a specific machine contact Mercury Interactive's Customer Support.

To modify the current license information:

1 Click **Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner** to open the Mercury LoadRunner window, and then click the **License** button. The LoadRunner License Information dialog box opens.

2 Click the **New License** button. The New LoadRunner License dialog box opens.



3 Enter the new license number exactly as it was given to you. Click **OK**. If your license is limited for a specific amount of time, LoadRunner issues a message accordingly.

4 Click **OK** to close the New LoadRunner License dialog.

Part II

Designing a Tuning Session

2

Tuning Session Overview

Mercury Tuning Console guides you through the process of tuning your system. This chapter describes a typical tuning workflow.

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 Mercury Tuning Console’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 27, “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. Mercury Tuning Console's monitors and analysis display your performance in real time. During the session execution, Mercury Tuning Console 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 244.

➤ 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 245.

➤ Scan Recommended Values—Use a Scan Recommended Values step to compare the current values on a machine or service with the recommended values in Mercury Tuning Console’s knowledge base. See “Scan Recommended Values Step” on page 253.

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 Mercury Tuning Console’s integrated tuners to make changes to system component properties from the Console machine. See Chapter 29, “Introducing Remote Tuning.”

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 Mercury Tuning Console and open or create a session.

This chapter discusses:

- Creating or Opening a Session

About Getting Started

A Mercury Tuning Console 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 Mercury Tuning Console's knowledge base. For more information about cartridges, see "Using Built-In Knowledge" on page 89.

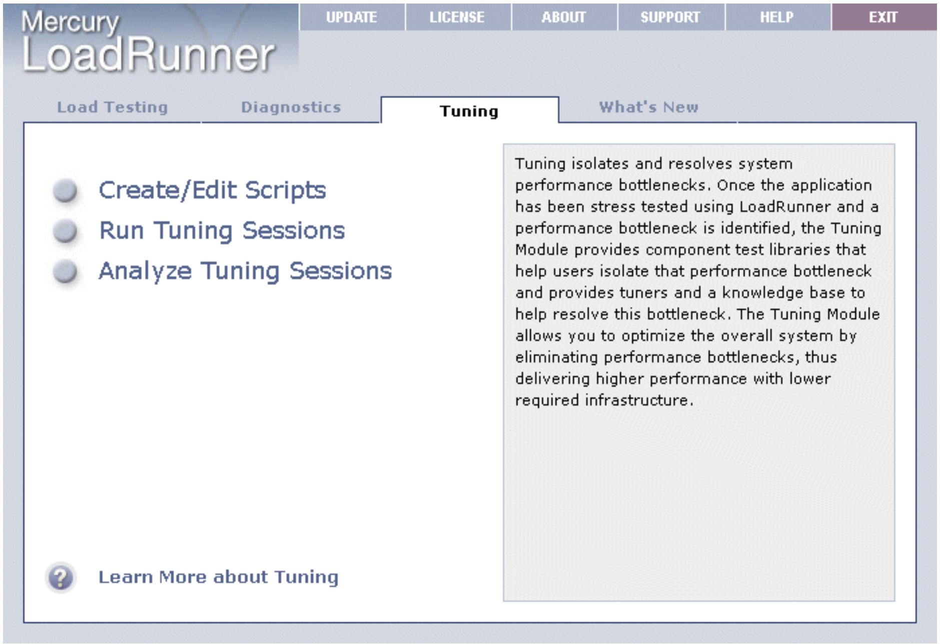
After choosing the cartridge, you then customize it to your specific environment.

Creating or Opening a Session

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

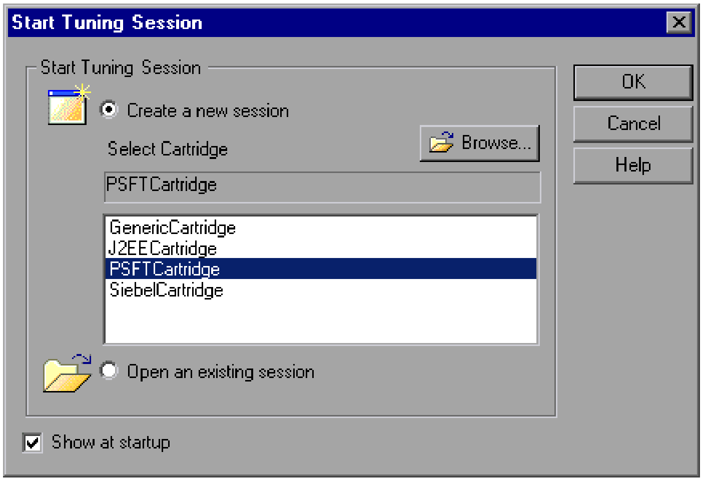
To create or open a Mercury Tuning Console session:

1 Choose **Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner**. Mercury Tuning Console displays the main Mercury LoadRunner screen:



Tuning isolates and resolves system performance bottlenecks. Once the application has been stress tested using LoadRunner and a performance bottleneck is identified, the Tuning Module provides component test libraries that help users isolate that performance bottleneck and provides tuners and a knowledge base to help resolve this bottleneck. The Tuning Module allows you to optimize the overall system by eliminating performance bottlenecks, thus delivering higher performance with lower required infrastructure.

2 Choose **Run Tuning Sessions**. The Start Tuning Session dialog box is displayed:



Start Tuning Session

Start Tuning Session

● Create a new session

Select Cartridge

PSFTCartridge

GenericCartridge
J2EECartridge
PSFTCartridge
SiebelCartridge

○ Open an existing session

☑ Show at startup

Note: Alternatively, instead of steps 1 and 2, you can choose **Start** > **Programs** > **Mercury LoadRunner** > **Applications** > **Tuning Console**.

3 Choose a cartridge from the displayed list or browse to locate one.

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

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

See “Viewing the Session Tab” on page 21.

Understanding the Start Tuning Session Dialog Box

The Start Tuning Session dialog box allows you to create a new session immediately when invoking Mercury Tuning Console, 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**, Mercury Tuning Console creates a new session based on the cartridge. For information about cartridges, see “Using Built-In Knowledge” on page 89.

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 Mercury Tuning Console 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

The Session tab contains the following sections:

- **Workflow:** 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.

- **Description:** Session or step description. When the session (the root element) is selected, this section explains what the session contains and how to use the various steps. When 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 29 for details.

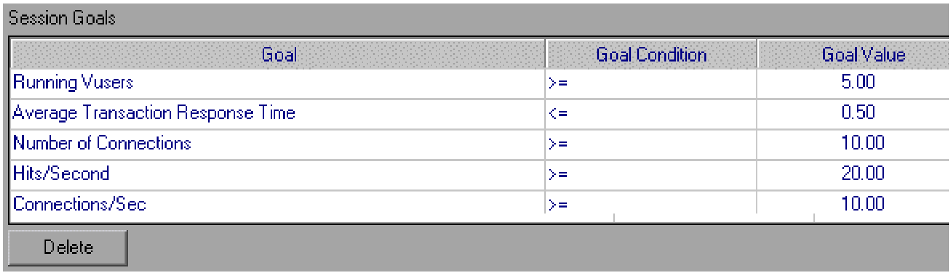
What you see in the other sections of the Session tab, and in the tool bar, depends on whether the session or a step is selected in the workflow section. In addition, if a step is selected, the display varies according to the type of step.

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:

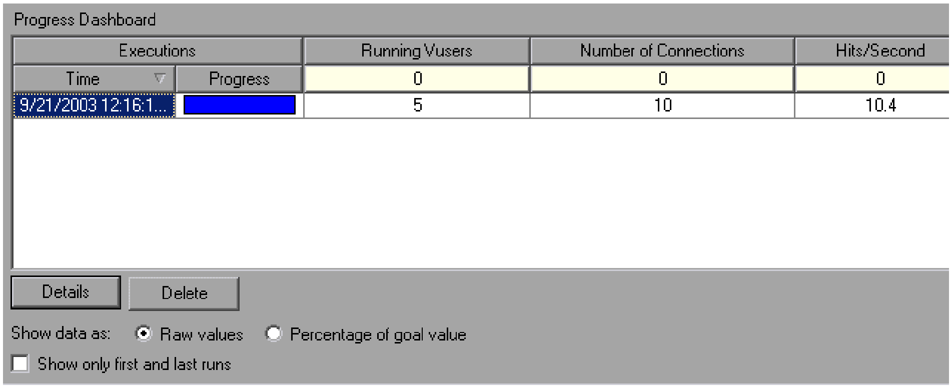


Session Goals

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. To delete a goal, select the relevant row and click **Delete**.

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



Progress Dashboard

Executions		Running Vusers	Number of Connections	Hits/Second
Time	Progress	0	0	0
9/21/2003 12:16:1...	■	5	10	10.4

Show data as: ● Raw values ○ Percentage of goal value

□ Show only first and last runs

See “Viewing Step Run Progress” on page 239 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 55.
Execution	“Creating and Configuring Execution Steps” on page 61.
Network Health Check	“Network Health Check Step” on page 244.
Scan Logs	“Scan Logs Step” on page 245.
Scan Recommended Values	“Scan Recommended Values Step” on page 253.

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 239.
- **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 52.
Remove Step	Removes a step from the session. See “Managing Steps” on page 53.
Edit Schedule Settings	Schedules the session step by defining a manual or goal-oriented profile. See “Scheduling Session Steps” on page 127.
Execute Session Step	Runs the selected Execution step. See “Running an Execution Step” on page 207.
Configure	Opens the Step Configuration dialog box. See “Understanding the Step Configuration Dialog Box” on page 74.
Stop Gradually	Gradually stops the Execution step. See “Running an Execution Step” on page 209.
Stop Immediately	Stops the Execution step immediately. See “Running an Execution Step” on page 209.
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 214.
View and Control Vusers	Opens the Vuser Management window. See “Understanding the Vuser Management Window” on page 218.
Add/Remove Graphs	Opens the Select Online Graphs dialog box. See “Understanding the Select Online Graphs Dialog Box” on page 359.

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 56.
Delete Task	Allows you to remove a task from a manual step. See “Creating a Manual Step” on page 56.
Move Task Up	Moves a task up in the table of a manual step. See “Creating a Manual Step” on page 56.
Move Task Down	Moves a task down in the table of a manual step. See “Creating a Manual Step” on page 56.
Select All Tasks	Selects all the tasks in the table of a manual step. See “Creating a Manual Step” on page 56.
Check Network Health	Executes a Network Health Check step. See “Network Health Check Step” on page 244.
Scan Recommended Values	Executes a Scan Recommended Values step. See “Scan Recommended Values Step” on page 253.
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 255.
Start Log Scan	Starts scanning log files for errors. See “Scanning Log Files” on page 246.
Stop Log Scan	Stops scanning log files. See “Scanning Log Files” on page 246.

▦	**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 246.
▽	**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 246.

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.

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

For a list of Mercury Tuning Console's topology templates, see "Understanding the Topology Templates Window" on page 35.

You can use one of these templates as is or customize it. When you add items to the topology, Mercury Tuning Console 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 Mercury Tuning Console's design window, you can drag topology elements from the design palette into the topology diagram workspace.

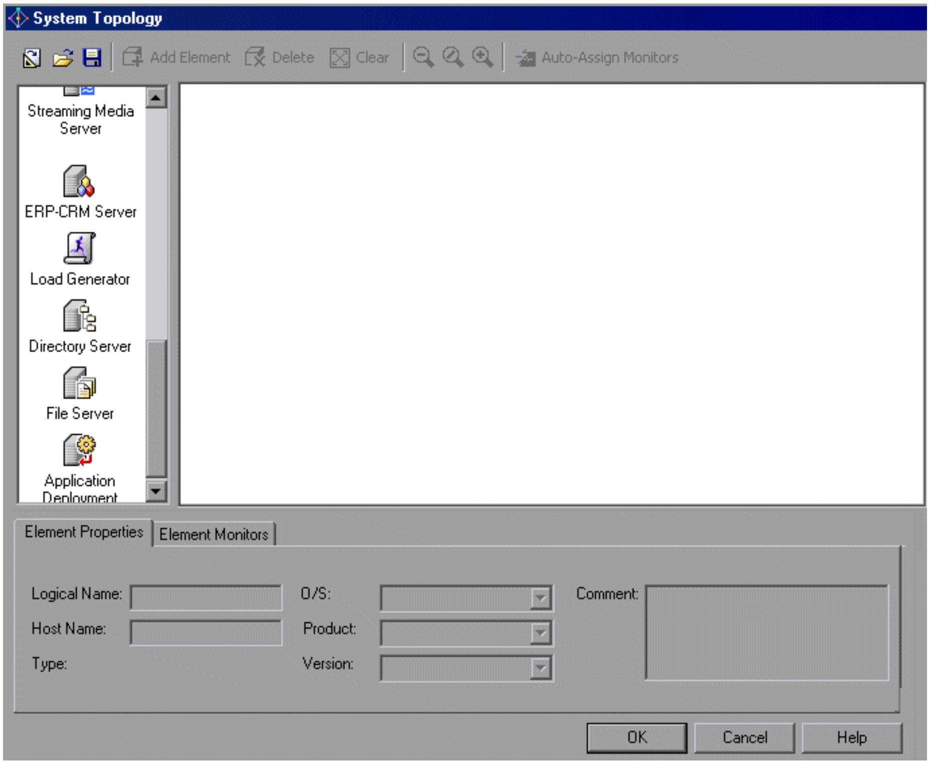
You can map a physical component to more than one logical topology element. For example, if a computer hosts both a Web server and a database server, you can define two elements—a Web server element and a database server element—and map them to the same computer.

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

To create a new topology diagram:

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 35). 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 40.

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 37.

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 48.

To import an existing topology diagram:

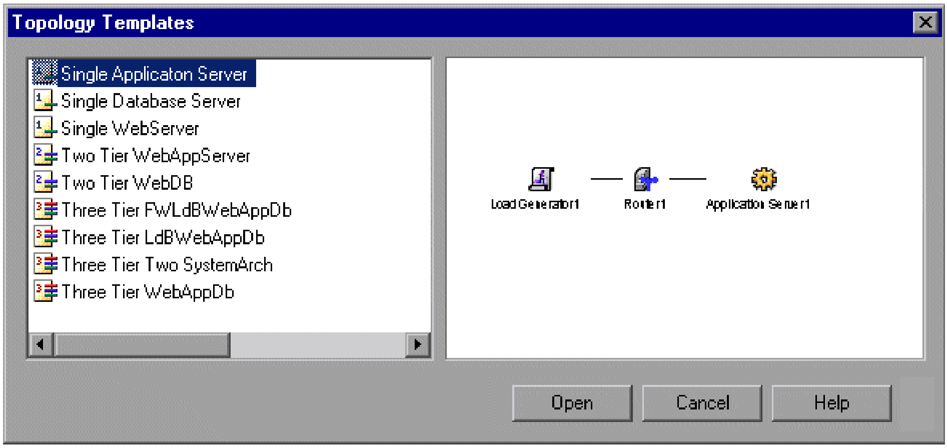
1 Click **Open Topology File** and browse to the desired topology (a file with a *.tpl* extension).

2 Click **Open**. Mercury Tuning Console 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.

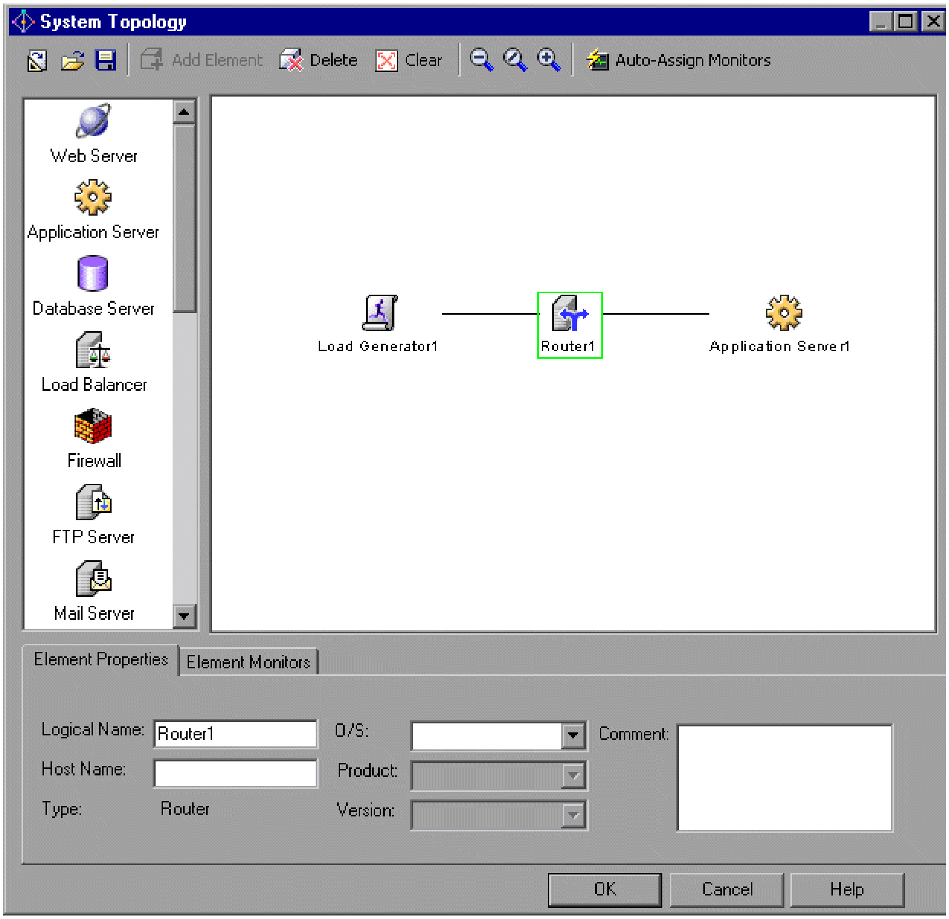


Topology Templates

- Single Applicaton Server
- Single Database Server
- Single WebServer
- Two Tier WebAppServer
- Two Tier WebDB
- Three Tier FWLdBWebAppDb
- Three Tier LdBWebAppDb
- Three Tier Two SystemArch
- Three Tier WebAppDb

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:



Logical Name: Router1
Host Name:
Type: Router

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 Mercury Tuning Console.

Following is a list of Mercury Tuning Console'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. Mercury Tuning Console builds a tuning session based on these settings.

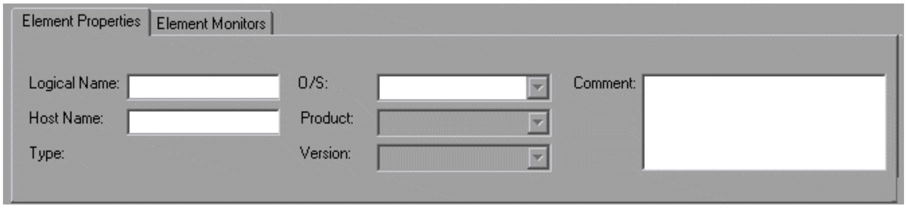
You can specify the following properties for each component:

- 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 Mercury Tuning Console 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.

Note: If you specify the component's logical name but not its host name, Mercury Tuning Console copies the logical name into the Host Name field. If you specify neither host name nor logical name, Mercury Tuning Console 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, Mercury Tuning Console 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.

This chapter discusses:

- Automatically Assigning Monitors and Alerts
- Manually Selecting Monitors
- 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

Mercury Tuning Console'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.

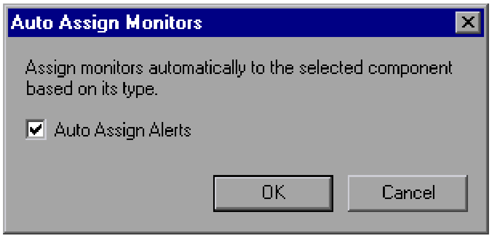
Tip: For details of Mercury Tuning Console's monitors, see "Monitoring a Session" on page 345.

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 Mercury Tuning Console's knowledge base. For information on alerts, see "Defining Alerts" on page 149.

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**. Mercury Tuning Console displays the Auto Assign Monitors dialog box:



Auto Assign Monitors

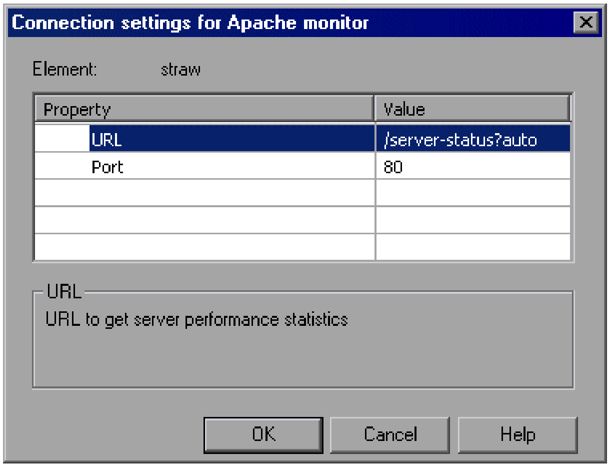
Assign monitors automatically to the selected component based on its type.

☑ Auto Assign Alerts

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, Mercury Tuning Console 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:



Connection settings for Apache monitor

Element: straw

Property	Value
URL	/server-status?auto
Port	80

URL

URL to get server performance statistics

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

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

   Mercury Tuning Console 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 Mercury Tuning Console 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

Manually Selecting Monitors

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

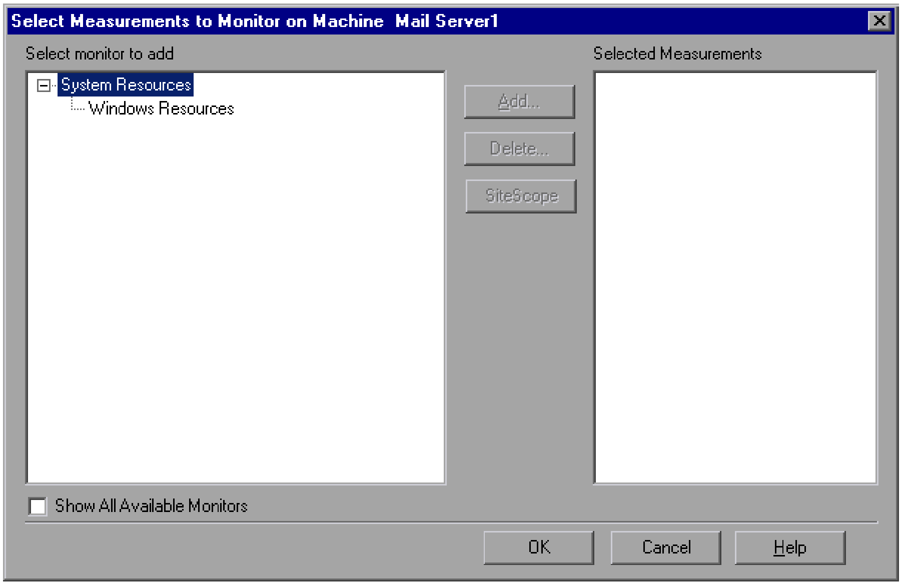
Note: You need to define the component properties before selecting the monitors, since Mercury Tuning Console uses these properties to locate the available monitors in its knowledge base.

To select a measurement to monitor:

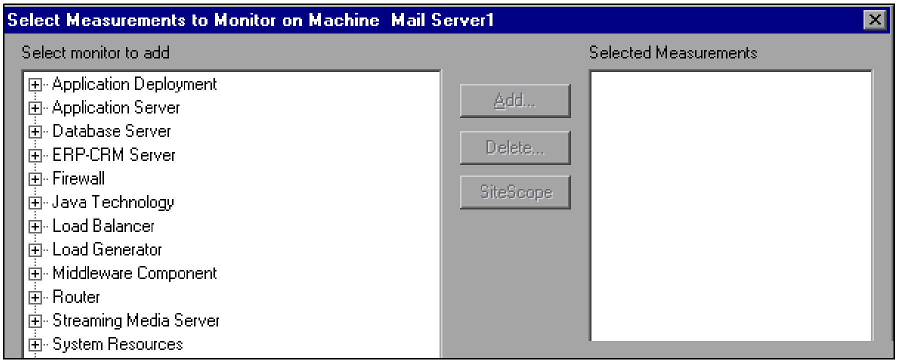
1. In the topology diagram, select the component whose properties you want to monitor.
2. Click the **Element Monitors** tab and then click **Add Monitor**. The **Select measurements to monitor** dialog box opens. For detailed information about this dialog box, see "Understanding the Select Measurements to Monitor Dialog Box" on page 48.

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 the Tuning Console" on page 353.

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



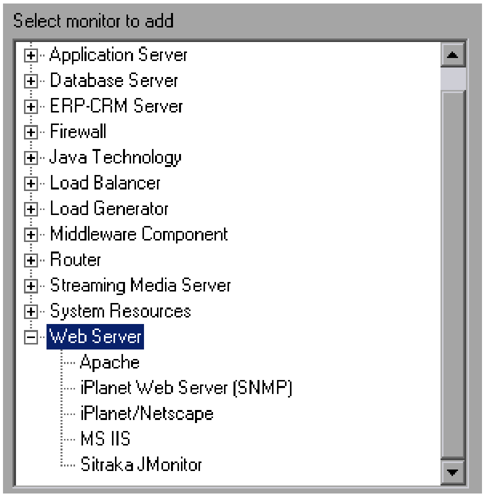
If the knowledge base does not contain a list of monitors for the specified element (possibly because you haven't specified all of the element's properties), Mercury Tuning Console displays all the monitors in its knowledge base:



If not all of Mercury Tuning Console'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.

3 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:

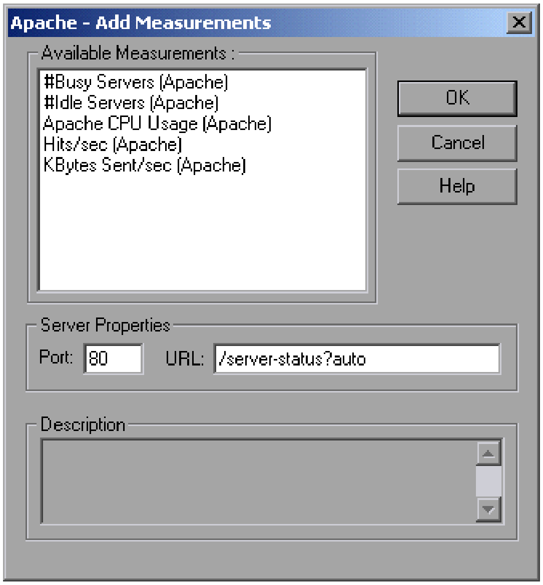


4 Click a component to select it.

If the monitor you have selected requires configuration of a SiteScope server, the SiteScope button becomes enabled. Click **SiteScope** to display the SiteScope Server Configuration dialog box, enter the required information, and click **OK**. The Select Measurements to Monitor dialog box is redisplayed.

5 Click **Add** or double-click the monitor. For most component types, Mercury Tuning Console opens a dialog box with a list of the component's available measurements.

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



Apache - Add Measurements

Available Measurements :

#Busy Servers (Apache)
#Idle Servers (Apache)
Apache CPU Usage (Apache)
Hits/sec (Apache)
KBytes Sent/sec (Apache)

Server Properties

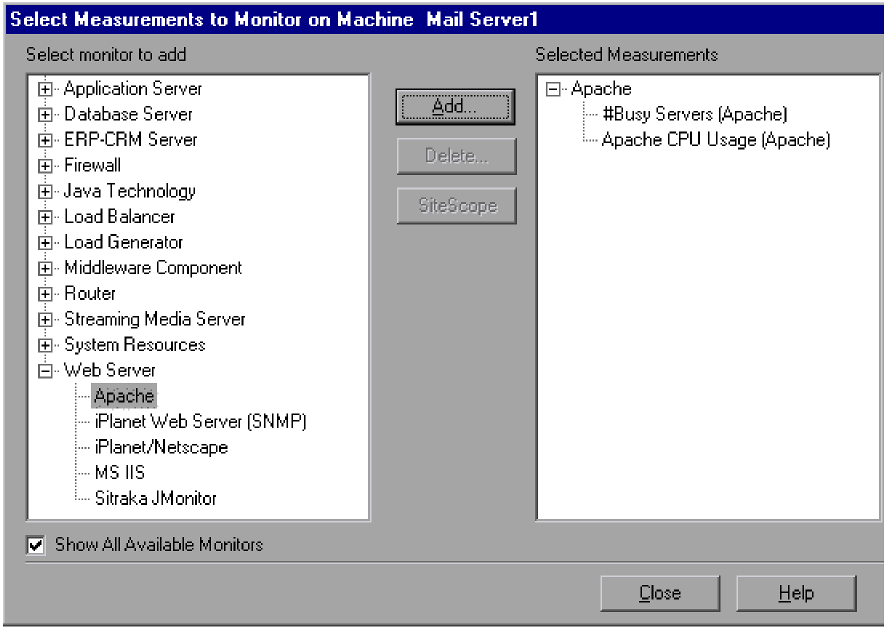
Port: 80 URL: /server-status?auto

Description

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 345.

---

6 Select the measurements that you want to monitor, and click **Add**. Mercury Tuning Console adds the measurements to the Selected Measurements pane.



Select Measurements to Monitor on Machine Mail Server1

Select monitor to add

- Application Server
- Database Server
- ERP-CRM Server
- Firewall
- Java Technology
- Load Balancer
- Load Generator
- Middleware Component
- Router
- Streaming Media Server
- System Resources
- Web Server
  - Apache
  - iPlanet Web Server (SNMP)
  - iPlanet/Netscape
  - MS IIS
  - Sitraka JMonitor

☑ Show All Available Monitors

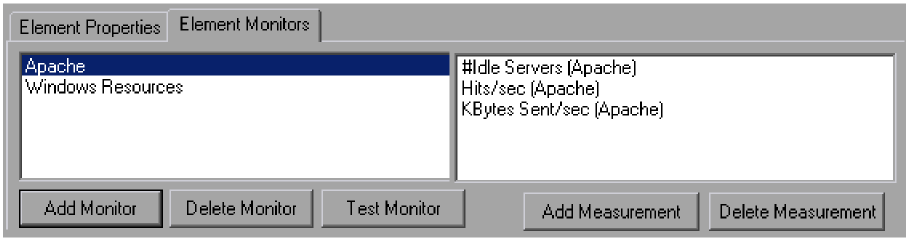
Selected Measurements

- Apache
  - #Busy Servers (Apache)
  - Apache CPU Usage (Apache)

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

8 Repeat steps 1 through 6 for all the components that you want to monitor, and then click **Close**.

Mercury Tuning Console displays the newly added monitors in the Element Monitors section.



Apache
Windows Resources

#Idle Servers (Apache)
Hits/sec (Apache)
KBytes Sent/sec (Apache)

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 Mercury Tuning Console 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 Mercury Tuning Console 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.

SiteScope: If the selected monitor requires configuration of a SiteScope server, this button is enabled. Click it to open the SiteScope Server Configuration dialog box.

Show All Available Monitors: Check this box to display a list of all of Mercury Tuning Console'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.

Understanding the SiteScope Server Configuration Dialog Box

You use the SiteScope Server Configuration dialog box to enter information about the SiteScope server required for some of the monitors.

Name: Host name of the SiteScope server computer

Port: The port used by the SiteScope server

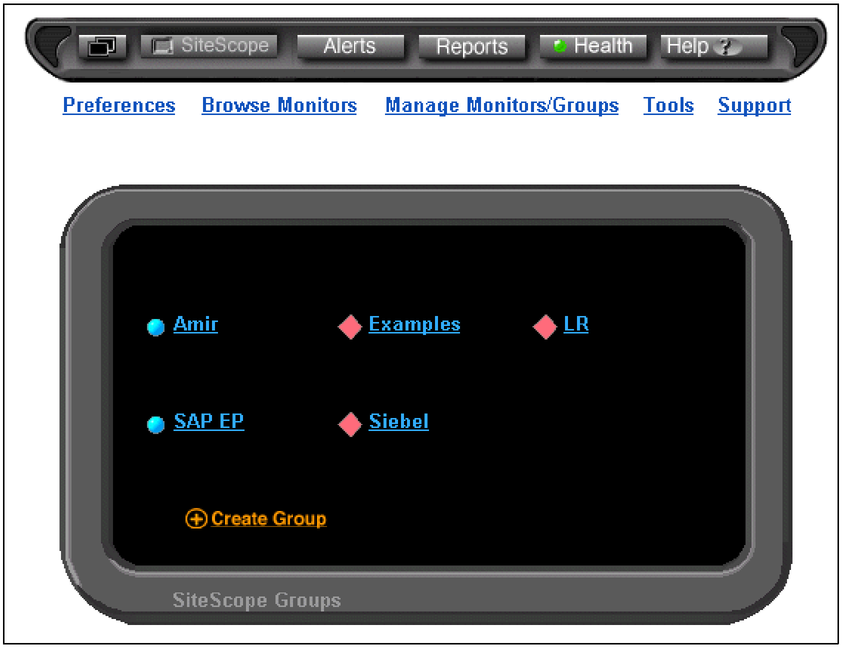
Version: SiteScope version

Accessing the SiteScope Administration Console

If you define a SiteScope Server element, Mercury Tuning Console 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 Mercury Tuning Console. After you perform each task, you check a box to indicate its completion. These steps guide you through tasks that Mercury Tuning Console 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 55.
   - **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 61.
   - **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 243.

- **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 245.
- **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 253.
- **Import LoadRunner Scenario:** Lets you create an execution step from an existing LoadRunner scenario. See “Importing LoadRunner Scenarios” on page 70.

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

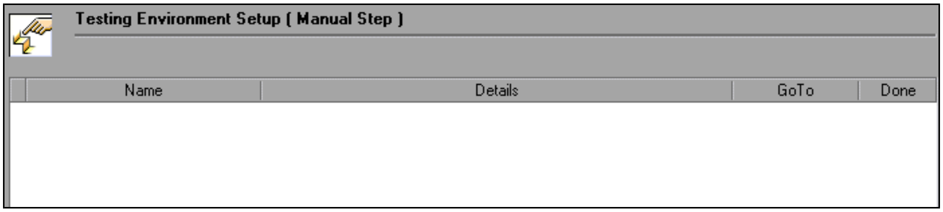
About Defining a Manual Step

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

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.



Testing Environment Setup (Manual Step)

Name	Details	GoTo	Done

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

Adding, Modifying and Deleting Tasks in a Manual Step

You can add tasks to a manual step, modify existing tasks, or delete tasks from the step.

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.

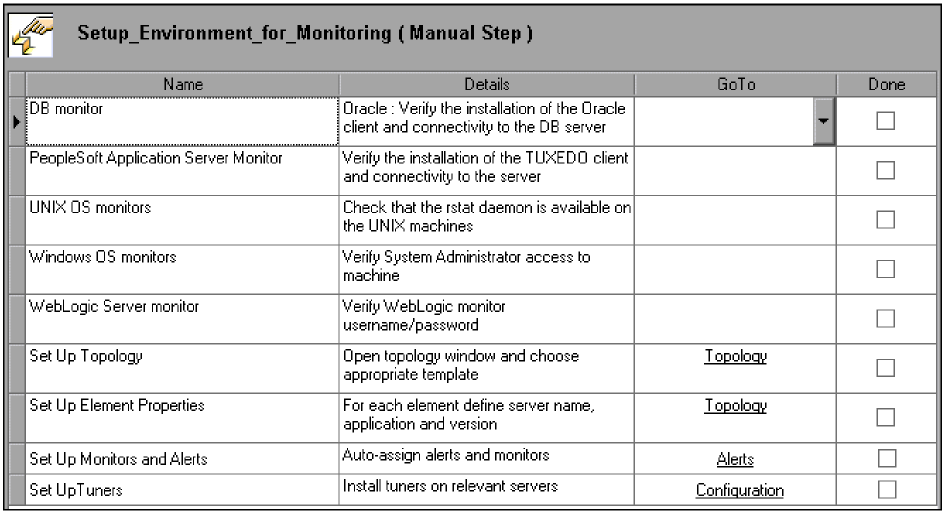
2. Click the **Name** field and enter the name of the task.
3. Click the **Details** field and enter an explanation of what the user needs to do.

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, Mercury Tuning Console 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 32.
Configuration	Configuration tab. See “Understanding the Configuration Tab,” on page 385.
Alerts	Alerts window. See “Understanding the Alerts Window,” on page 160.
Monitors	Monitor Configuration dialog box. See “Understanding the Monitors Configuration Dialog Box,” on page 355.

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:



Setup_Environment_for_Monitoring (Manual Step)

Name	Details	GoTo	Done
DB monitor	Oracle : Verify the installation of the Oracle client and connectivity to the DB server		□
PeopleSoft Application Server Monitor	Verify the installation of the TUXEDO client and connectivity to the server		□
UNIX OS monitors	Check that the rstat daemon is available on the UNIX machines		□
Windows OS monitors	Verify System Administrator access to machine		□
WebLogic Server monitor	Verify WebLogic monitor username/password		□
Set Up Topology	Open topology window and choose appropriate template	Topology	□
Set Up Element Properties	For each element define server name, application and version	Topology	□
Set Up Monitors and Alerts	Auto-assign alerts and monitors	Alerts	□
Set UpTuners	Install tuners on relevant servers	Configuration	□

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**.

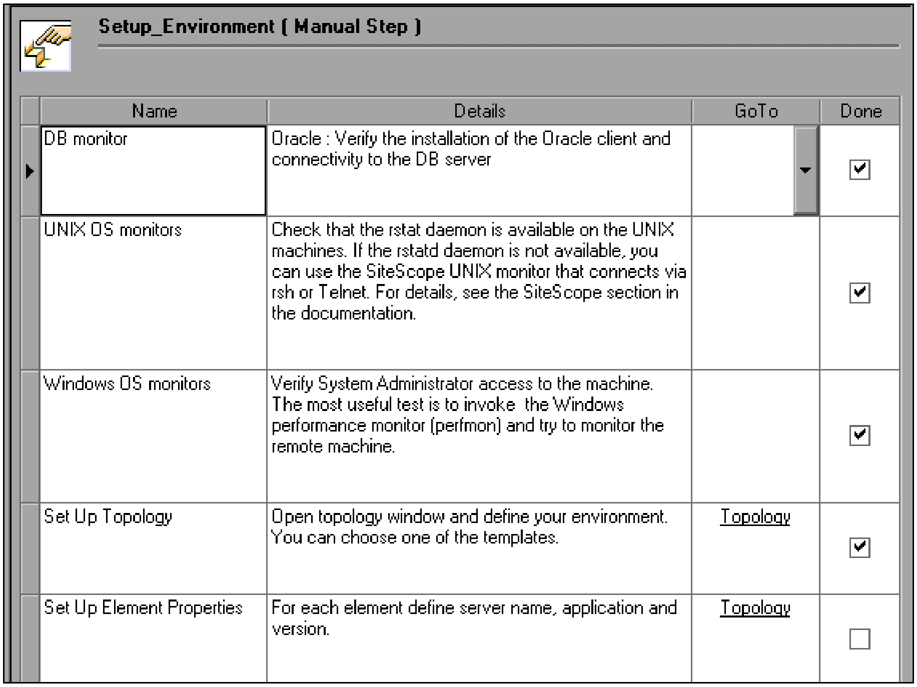
2 To delete, click **Delete Task**. Alternatively, you can right-click the task and choose **Delete Tasks.**

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. Mercury Tuning Console 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 97.

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 Mercury Tuning Console executes single steps, combining scripts in a step causes them to be executed together.

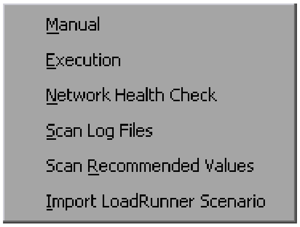
Vusers are virtual users that emulate real users. After setting up the steps for your session, you can indicate the number of Vusers you want to emulate. You can also choose a goal for each session step. For more information, see Chapter 12, "Scheduling Session Steps."

Creating New Execution Session Steps

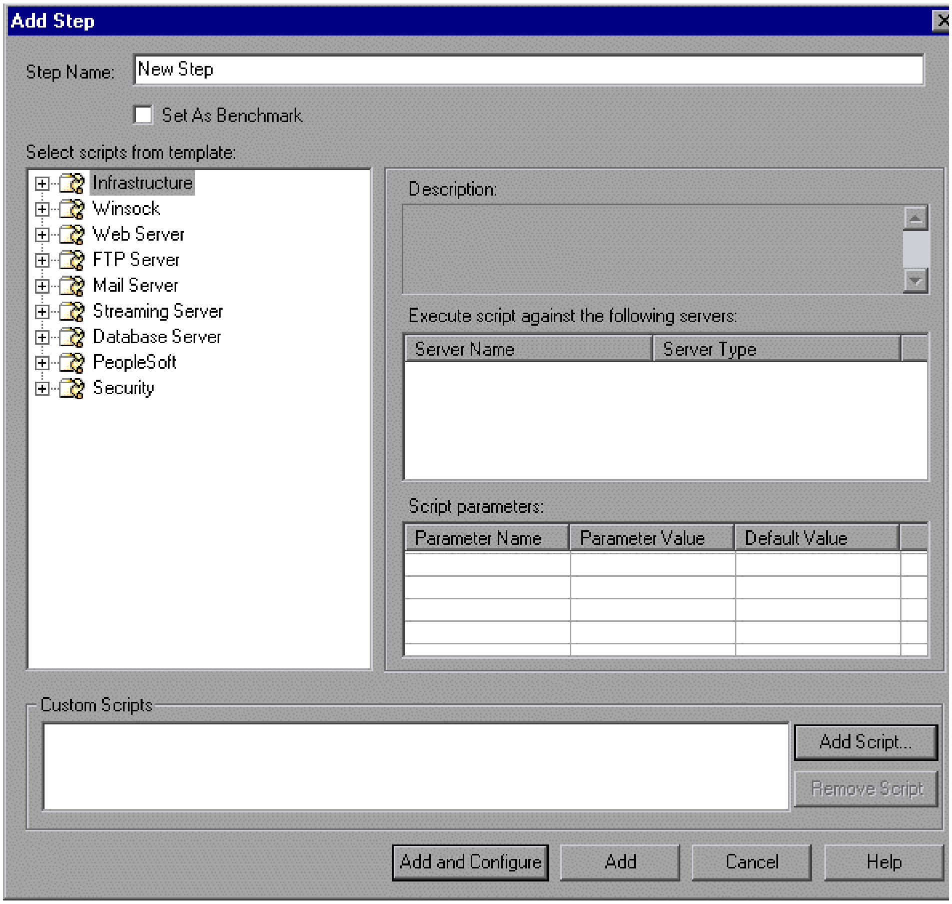
Creating a new Execution step involves choosing and configuring scripts.

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.



Add Step

Step Name: New Step

□ Set As Benchmark

Select scripts from template:

- Infrastructure
- Winsock
- Web Server
- FTP Server
- Mail Server
- Streaming Server
- Database Server
- PeopleSoft
- Security

Description:

Execute script against the following servers:

Server Name	Server Type

Script parameters:

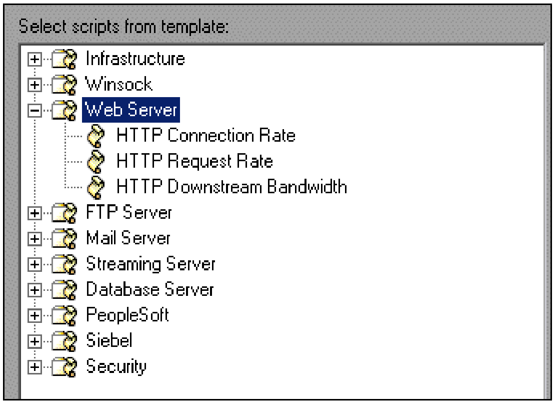
Parameter Name	Parameter Value	Default Value

Custom Scripts

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, choose **Set As Benchmark**. The benchmark step is used in the Progress Dashboard to measure and display improvements in performance. See “Viewing the Progress Dashboard,” on page 240.

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.

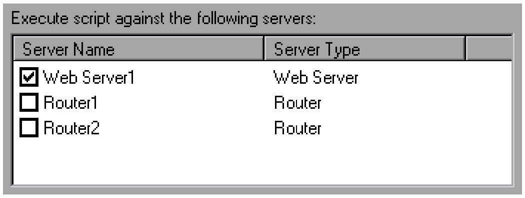


Select scripts from template:

- Infrastructure
- Winsock
- Web Server
  - HTTP Connection Rate
  - HTTP Request Rate
  - HTTP Downstream Bandwidth
- FTP Server
- Mail Server
- Streaming Server
- Database Server
- PeopleSoft
- Siebel
- Security

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

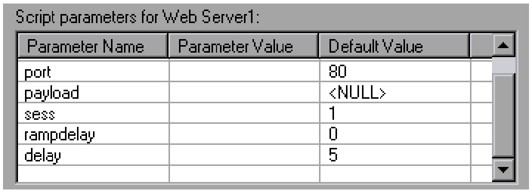
7 In the **Execute script against the following servers** section, Mercury Tuning Console displays the servers against which you can run the script. To specify the servers against which you want the script to run, check the boxes adjacent to their names.



Execute script against the following servers:

Server Name	Server Type
☑ Web Server1	Web Server
□ Router1	Router
□ Router2	Router

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



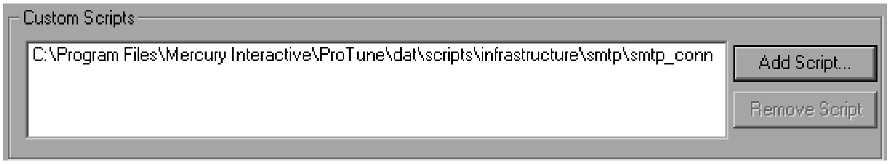
Script parameters for Web Server1:

Parameter Name	Parameter Value	Default Value
port		80
payload		<NULL>
sess		1
rampdelay		0
delay		5

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 Script** button, browse to the script that you want to use, and click **Open**. The script name is added to the Custom Scripts pane.



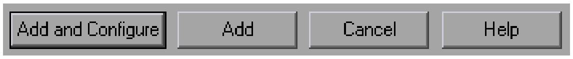
Custom Scripts

C:\Program Files\Mercury Interactive\ProTune\dat\scripts\infrastructure\smtp\smtp_conn

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 is used in the Progress Dashboard to measure and display improvements in performance (see “Viewing the Progress Dashboard,” on page 240). 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 Script**: 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 Script:** Removes a selected custom script from the Custom Scripts pane.
- **Add:** 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.

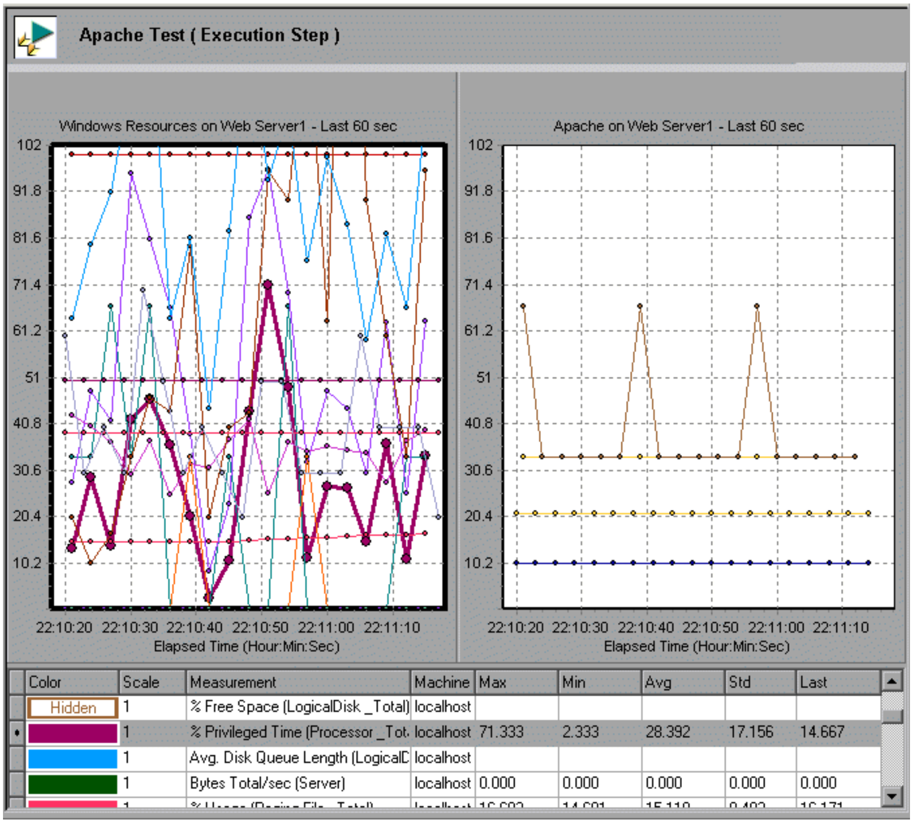
Description: The name and description of the selected script.

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.


Apache Test (Execution Step)
Windows Resources on Web Server1 - Last 60 sec
Apache on Web Server1 - Last 60 sec
102
91.8
81.6
71.4
61.2
51
40.8
30.6
20.4
10.2
22:10:20 22:10:30 22:10:40 22:10:50 22:11:00 22:11:10
Elapsed Time (Hour:Min:Sec)

	Color	Scale	Measurement	Machine	Max	Min	Avg	Std	Last
	Hidden	1	% Free Space (LogicalDisk _Total)	localhost					
•	🟪	1	% Privileged Time (Processor _Tot	localhost	71.333	2.333	28.392	17.156	14.667
	🟦	1	Avg. Disk Queue Length (LogicalD	localhost					
	🟩	1	Bytes Total/sec (Server)	localhost	0.000	0.000	0.000	0.000	0.000
	🟥	1	% Usage (Paging File _Total)	localhost	16.603	14.691	15.110	0.493	16.171

The tool bar includes the following buttons for controlling and configuring the step:

Remove Step: Deletes the selected step from the session.

Configure: Opens the Step Configuration dialog box. See "Understanding the Step Configuration Dialog Box," on page 74.

Scheduler: Opens the Schedule Builder window. See "Scheduling Session Steps," on page 127.

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 359.

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.

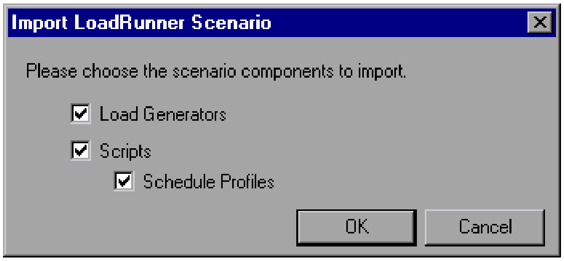
Note: You can import goal-oriented scenarios and manual scenarios in the Percentage Mode. If you need to import a scenario created in the Vuser Group Mode, first convert it (using LoadRunner Controller) to the Percentage Mode (**Scenario** > **Convert Scenario to the Percentage Mode**) and then import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



Import LoadRunner Scenario

Please choose the scenario components to import.

☑ Load Generators

☑ Scripts

☑ Schedule Profiles

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

Mercury Tuning Console 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.

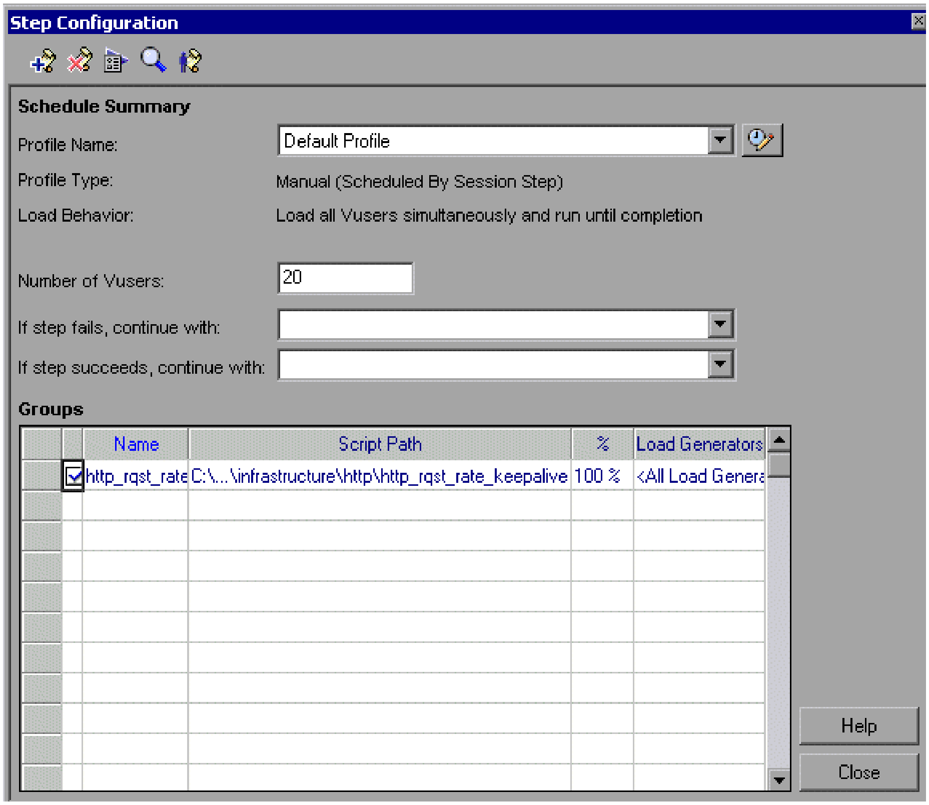
For the purpose of determining which step should subsequently execute, a failed step is one that ends when an alert causes all the Vusers to halt execution. If the step ends in any other way, it is considered to be a successful step.

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 63).

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



Step Configuration

Schedule Summary

Profile Name: Default Profile

Profile Type: Manual (Scheduled By Session Step)

Load Behavior: Load all Vusers simultaneously and run until completion

Number of Vusers: 20

If step fails, continue with:

If step succeeds, continue with:

Groups

		Name	Script Path	%	Load Generators
	☑	http_rqst_rate	C:\...\infrastructure\http\http_rqst_rate_keepalive	100 %	<All Load Genera

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 *Mercury Virtual User Generator User's Guide*.

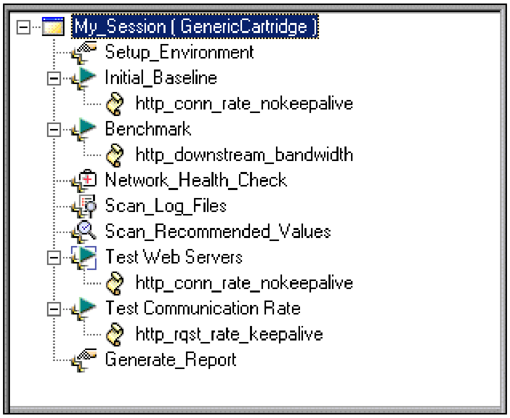
The Schedule Summary section includes the following fields:

- **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 146.
- **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 130.

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.



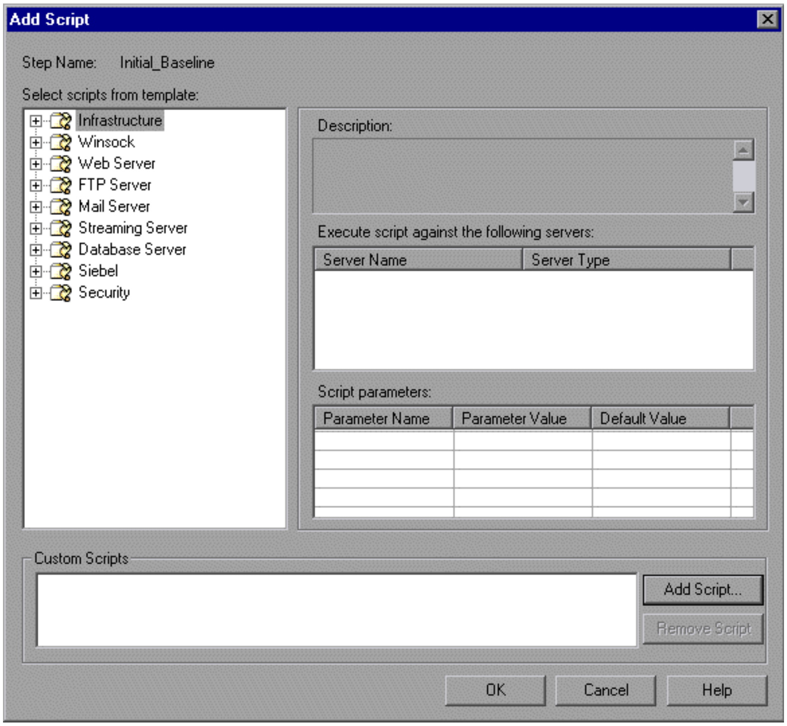
Adding a Script to a Step

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.



Add Script

Step Name: Initial_Baseline

Select scripts from template:

- Infrastructure
- Winsock
- Web Server
- FTP Server
- Mail Server
- Streaming Server
- Database Server
- Siebel
- Security

Description:

Execute script against the following servers:

Server Name	Server Type

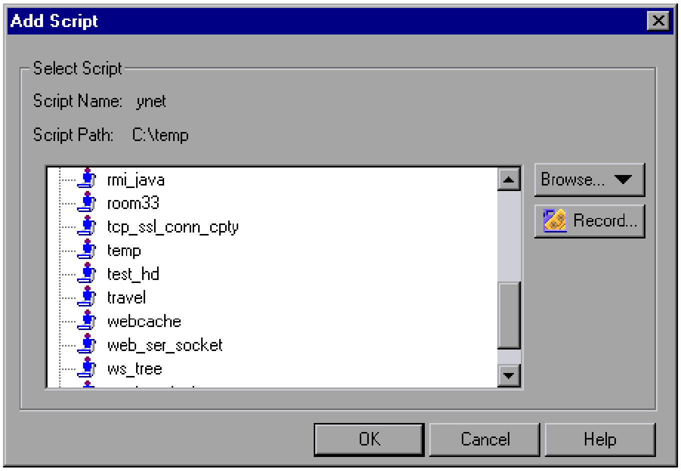
Script parameters:

Parameter Name	Parameter Value	Default Value

Custom Scripts

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 63.

4 To add a custom script to your step, click **Add Script** 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 Mercury Tuning Console Virtual User Generator, click **Record**. For more information about recording scripts, refer to the *Mercury Virtual User Generator User's 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 *Mercury Virtual User Generator User's 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**. Mercury Tuning Console opens the Virtual User Generator, so that you can edit the script. For more information on editing scripts, refer to the *Mercury Virtual User Generator User's Guide*.

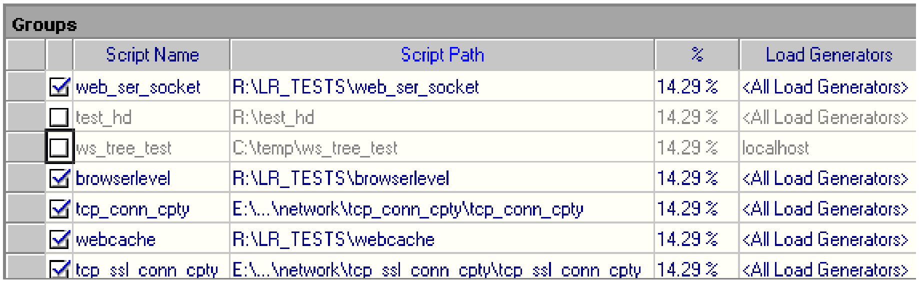
Working with Groups

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, Mercury Tuning Console runs them simultaneously. By default, Mercury Tuning Console 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:

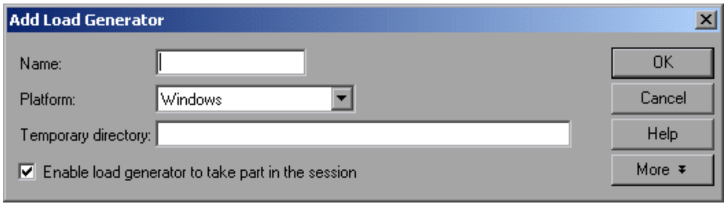
1 The Load Generators column automatically contains <All Load Generators> for each script. You can assign specific load generators for each script. Click in the **Load Generators** column in the next available row.

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 Mercury Tuning Console 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:



Add Load Generator

Name:

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

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, Mercury Tuning Console 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. Mercury Tuning Console 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 109. To configure additional load generator settings, see "Configuring Load Generator Settings" on page 115.

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

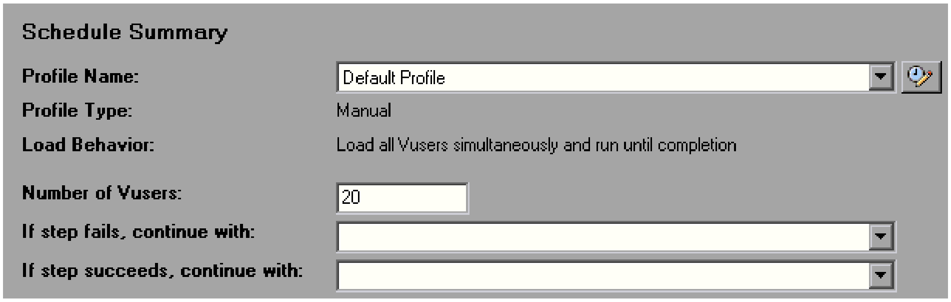
Setting an Initial Load (Manual Profiles)

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



Schedule Summary

Profile Name: Default Profile

Profile Type: Manual

Load Behavior: Load all Vusers simultaneously and run until completion

Number of Vusers: 20

If step fails, continue with:

If step succeeds, continue with:

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**)—Mercury Tuning Console runs the session step at the designated time with the specified number of Vusers.

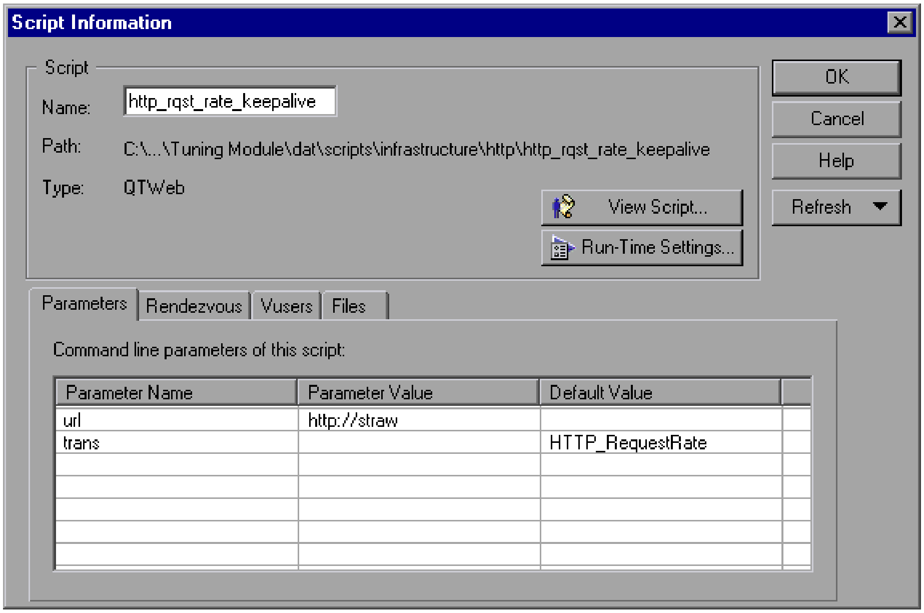
Note that you can change the number of Vusers while a step is executing, via the Run/Stop Vusers dialog box. See “Adding Vusers to a Running Execution Step” on page 214.

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.



Script Information

Script

Name: http_rqst_rate_keepalive

Path: C:\...\Tuning Module\dat\scripts\infrastructure\http\http_rqst_rate_keepalive

Type: QTWeb

Command line parameters of this script:

Parameter Name	Parameter Value	Default Value
url	http://straw	
trans		HTTP_RequestRate

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 13 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 18, "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 *Mercury Virtual User Generator User's 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 *Mercury Virtual User Generator User's Guide*.

Note: If you modify the run-time settings from the Console, Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console installation directory by typing a percent sign (%) at the beginning of the script path. For example, if the Mercury Tuning Console installation directory is located at *F:\Mercury Tuning Console*, to specify a script located at *F:\Mercury Tuning\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 33.
- **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 52.
- **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 “Automatically Assigning Monitors and Alerts” on page 40.

- **Settings for auto-assigned alerts:** The alerts that are assigned when you use the Auto Assign feature. See “Automatically Assigning Monitors and Alerts” on page 40.
- **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 420.
- **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 245.
- **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.

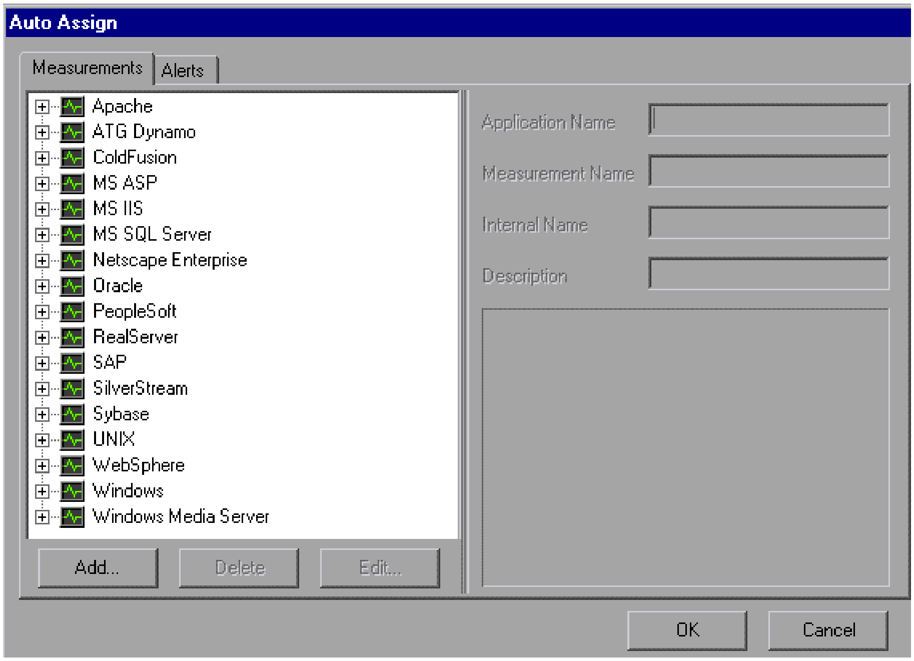
Editing Auto Assign Settings

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.

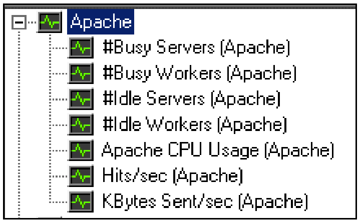
3 Use the Alerts tab to configure alerts.

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.



Apache
#Busy Servers (Apache)
#Busy Workers (Apache)
#Idle Servers (Apache)
#Idle Workers (Apache)
Apache CPU Usage (Apache)
Hits/sec (Apache)
KBytes Sent/sec (Apache)

Add: Click to add a measurement to the monitor list. See "Adding Measurements" on page 94.

Delete: Click to delete a measurement from the monitor list. See "Deleting Measurements" on page 96.

Alerts

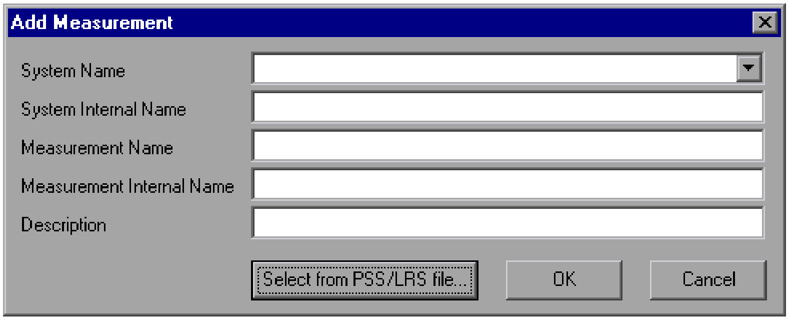
The Alerts tab displays all the available alerts, the conditions that trigger them, the actions they cause, and the relevant troubleshooting information. You can change the list of alerts that are applied by the Auto Assign feature. See "Editing Alerts" on page 96.

Adding Measurements

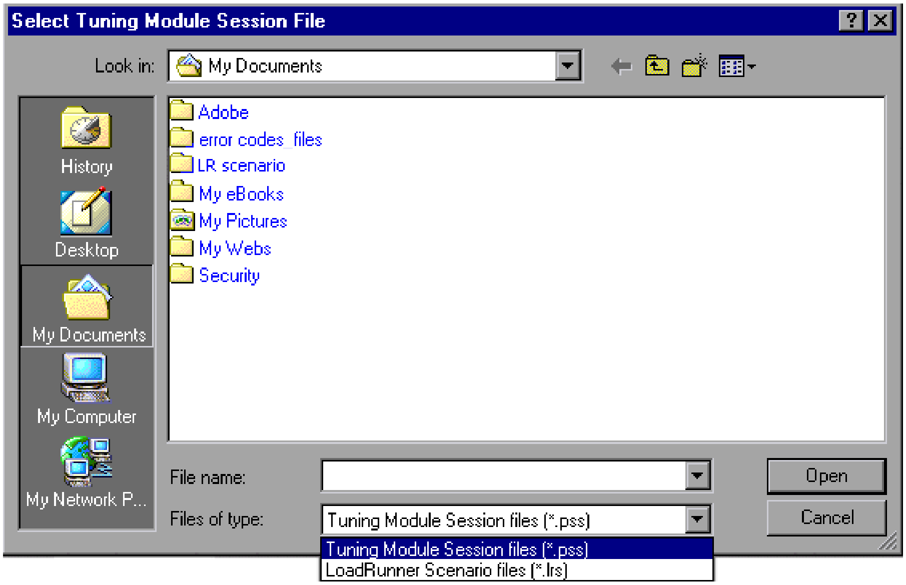
You use the Measurements tab to add auto-assigned measurements to monitors.

To add a measurement to a monitor:

1 Click **Add**. The Add Measurement dialog box is displayed.



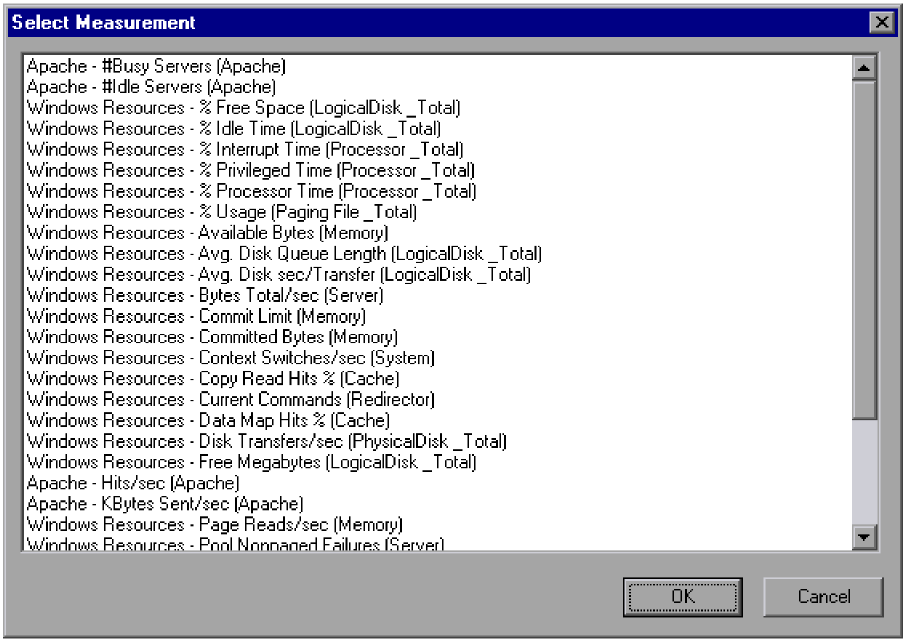
2 Click **Select from PSS/LRS file**. The Select Tuning Console Session File dialog box is displayed.


Look in: My Documents

Files of type: Tuning Module Session files (*.pss)

3 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).



Apache - #Busy Servers (Apache)
Apache - #Idle Servers (Apache)
Windows Resources - % Free Space (LogicalDisk _Total)
Windows Resources - % Idle Time (LogicalDisk _Total)
Windows Resources - % Interrupt Time (Processor _Total)
Windows Resources - % Privileged Time (Processor _Total)
Windows Resources - % Processor Time (Processor _Total)
Windows Resources - % Usage (Paging File _Total)
Windows Resources - Available Bytes (Memory)
Windows Resources - Avg. Disk Queue Length (LogicalDisk _Total)
Windows Resources - Avg. Disk sec/Transfer (LogicalDisk _Total)
Windows Resources - Bytes Total/sec (Server)
Windows Resources - Commit Limit (Memory)
Windows Resources - Committed Bytes (Memory)
Windows Resources - Context Switches/sec (System)
Windows Resources - Copy Read Hits % (Cache)
Windows Resources - Current Commands (Redirector)
Windows Resources - Data Map Hits % (Cache)
Windows Resources - Disk Transfers/sec (PhysicalDisk _Total)
Windows Resources - Free Megabytes (LogicalDisk _Total)
Apache - Hits/sec (Apache)
Apache - KBytes Sent/sec (Apache)
Windows Resources - Page Reads/sec (Memory)
Windows Resources - Pool Nonpaged Failures (Server)

4 Select the measurement that you want to add, and click **OK**. The Add Measurement dialog box is redisplayed. Note that the fields are now filled with the measurement's values.

5 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 make changes to the list of automatically-assigned alerts, click the Alerts tab to add, delete, or configure alerts.

To add an alert to the list:

➤ Follow the procedure described in "Creating an Alert," on page 157, starting with step 2. The alert is added to the list in the Available Alerts section and will be assigned when you use the Auto Assign feature.

If the alert already appears on the Available Alerts list, but its adjacent check box is cleared, check the box to cause the Tuning Console to assign it when the Auto Assign feature is used.

To delete an alert from the list:

➤ Select the alert in the Available Alerts pane and click **Delete**. The alert is removed from the list in the Available Alerts section.

Alternatively, you can clear the box adjacent to the alert in the Available Alerts section. This prevents the alert from being assigned by the Auto Assign feature, but leaves it on the list for future use.

To configure an alert:

➤ See "Defining Alerts," on page 149.

Canned Scripts

Mercury Tuning Console'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 Mercury Tuning Console. 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, refer to the chapter "Security Monitoring" in the *LoadRunner Monitor Reference*.

Note: Before you can use a LoadRunner load generator (as opposed to a Mercury Tuning Console 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 480.

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 Mercury Tuning Console 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.

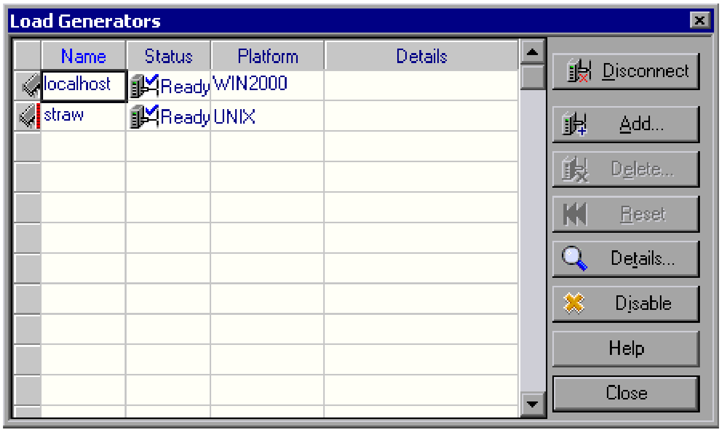
To configure global settings for all load generators participating in the session, use the Timeout tab in the **Tools** > **Options** dialog box. For more information, see Chapter 14, "Configuring Session Steps." To set properties specific to each load generator, use the Load Generators dialog box as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



Name	Status	Platform	Details
localhost	Ready	WIN2000	
straw	Ready	UNIX	

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 115.

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.

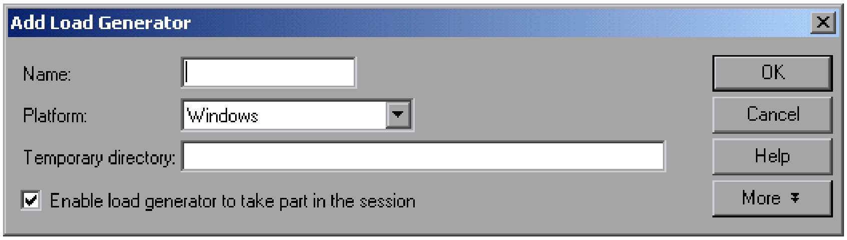
Note: The Console monitors a Windows load generator machine's CPU usage and automatically stops loading Vusers on a load generator when it becomes overloaded. You can monitor the status of a machine's CPU usage using the icons in this dialog box. When the CPU usage of a load generator becomes problematic,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



Add Load Generator

Name:

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

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 Mercury Tuning 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, Mercury Tuning Console 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 115.

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 493.
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.

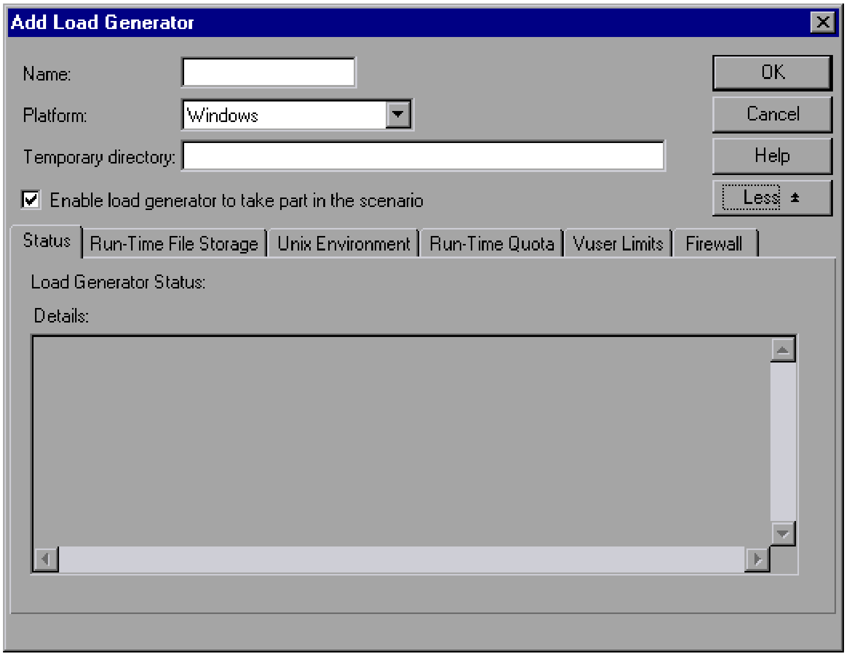
Platform: Select the type of platform on which the load generator is running.

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



Add Load Generator

Name:

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the scenario

Load Generator Status:

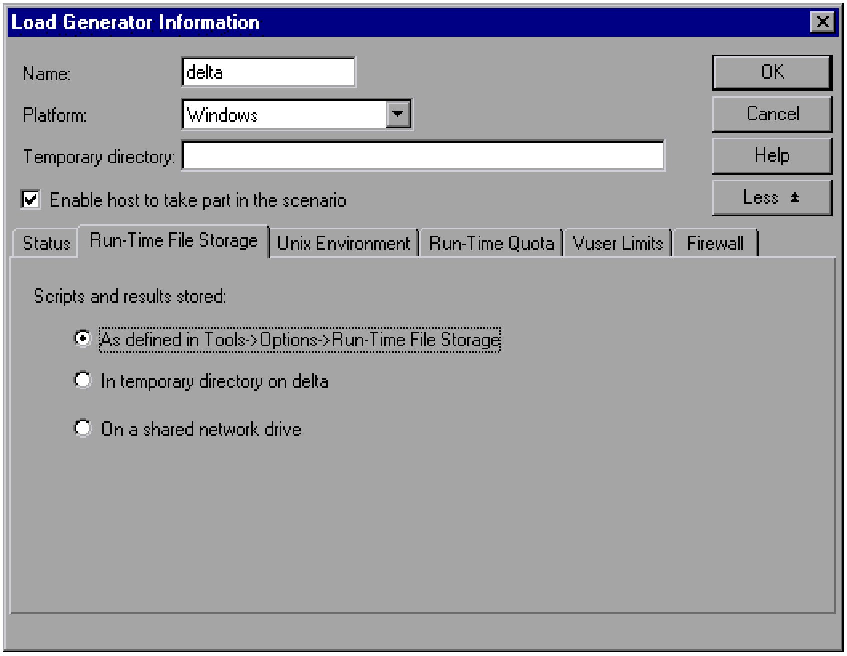
Details:

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 Mercury Tuning Console gathers from each load generator during a session.



Load Generator Information

Name: delta

Platform: Windows

Temporary directory:

☑ Enable host to take part in the scenario

Scripts and results stored:

● As defined in Tools->Options->Run-Time File Storage

○ In temporary directory on delta

○ On a shared network drive

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.

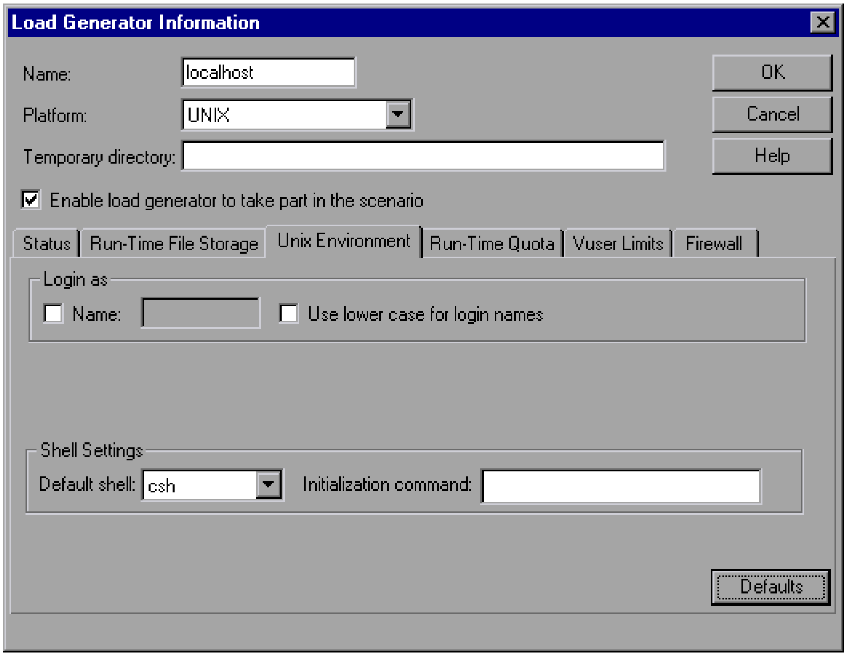
Note: If the load generator is *localhost*, Mercury Tuning Console stores the scripts and results on a shared network drive, and the checkboxes and radio buttons for setting the location are all disabled.

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.



Name: localhost

Platform: UNIX

Temporary directory:

☑ Enable load generator to take part in the scenario

Login as

☐ Name: ☐ Use lower case for login names

Shell Settings

Default shell: csh Initialization command:

Login as

- **Name:** If the load generator is UNIX-based, set the login information for the load generator. By default, Mercury Tuning Console 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 Mercury Tuning Console 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 493.

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 Mercury Tuning Console to run your application under the Korn shell, you first need to make sure that the *.profile* file contains all of the Mercury Tuning Console 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 Mercury Tuning Console environment settings.

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

➤ **Initialization command:** Enter any command line options for Mercury Tuning Console 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:

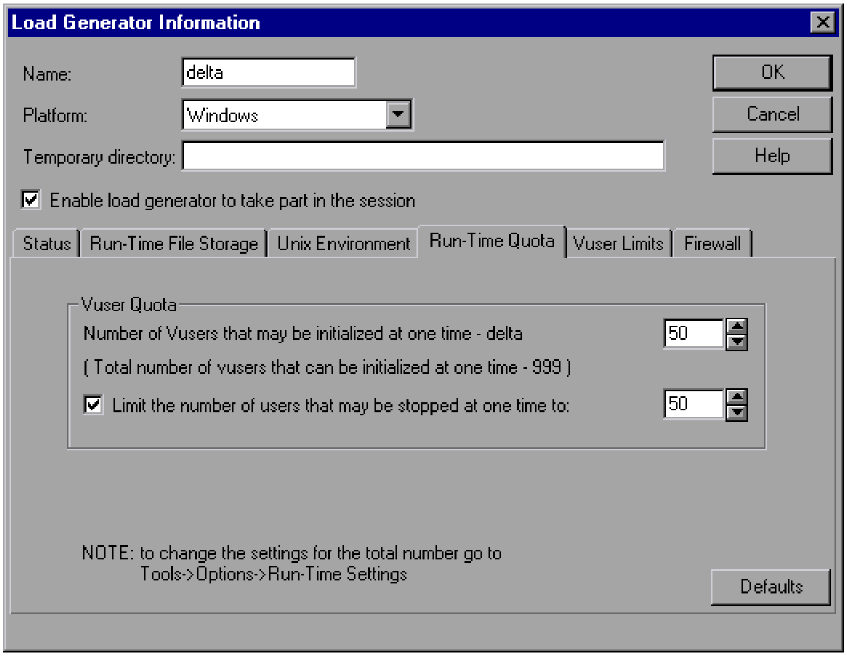
  ```
  . .profile;
  ```

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.



Load Generator Information

Name: delta

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

Vuser Quota

Number of Vusers that may be initialized at one time - delta: 50

(Total number of vusers that can be initialized at one time - 999)

☑ Limit the number of users that may be stopped at one time to: 50

NOTE: to change the settings for the total number go to Tools->Options->Run-Time Settings

Vuser Quota

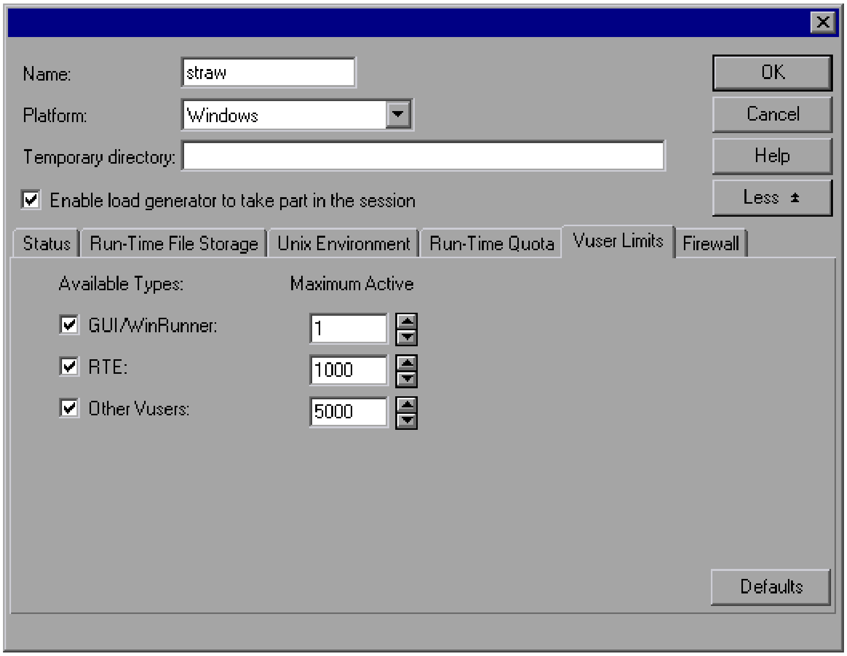
- **Number of Vusers that may be initialized at one time - current load generator:** Select the maximum number of Vusers that the current load generator can initialize simultaneously.
- **Limit the number of users that may be stopped at one time to:** Select the maximum number of Vusers that the current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



Name: straw

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

Available Types: / Maximum Active

☑ GUI/WinRunner: 1

☑ RTE: 1000

☑ Other Vusers: 5000

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

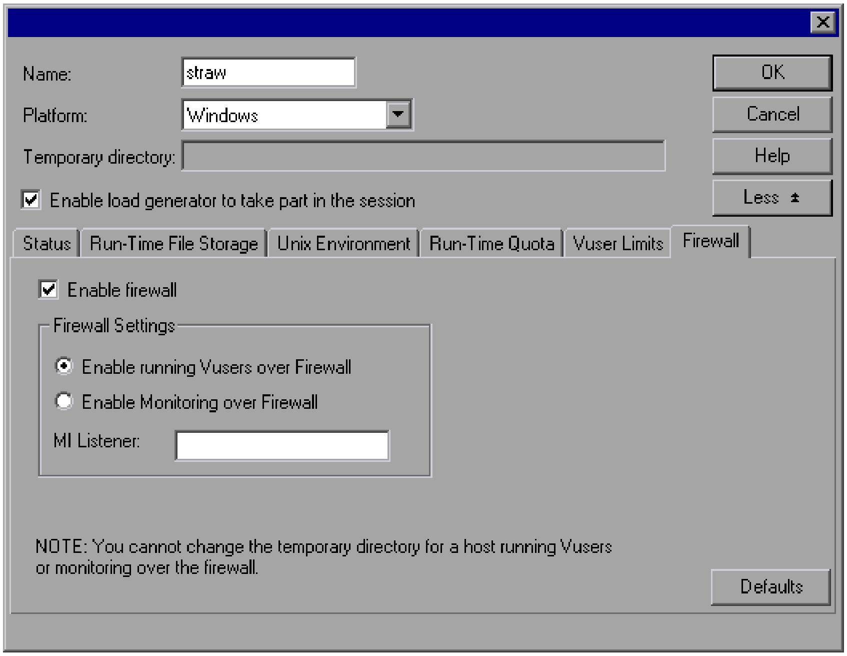
Maximum Active: Select the maximum number of Vusers of each type for the load generator to run.

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

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, in the Mercury LoadRunner window (**Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner**), click **License**.

Firewall Tab

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



Name: straw

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

☑ Enable firewall

Firewall Settings

● Enable running Vusers over Firewall

○ Enable Monitoring over Firewall

MI Listener:

NOTE: You cannot change the temporary directory for a host running Vusers or monitoring over the firewall.

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

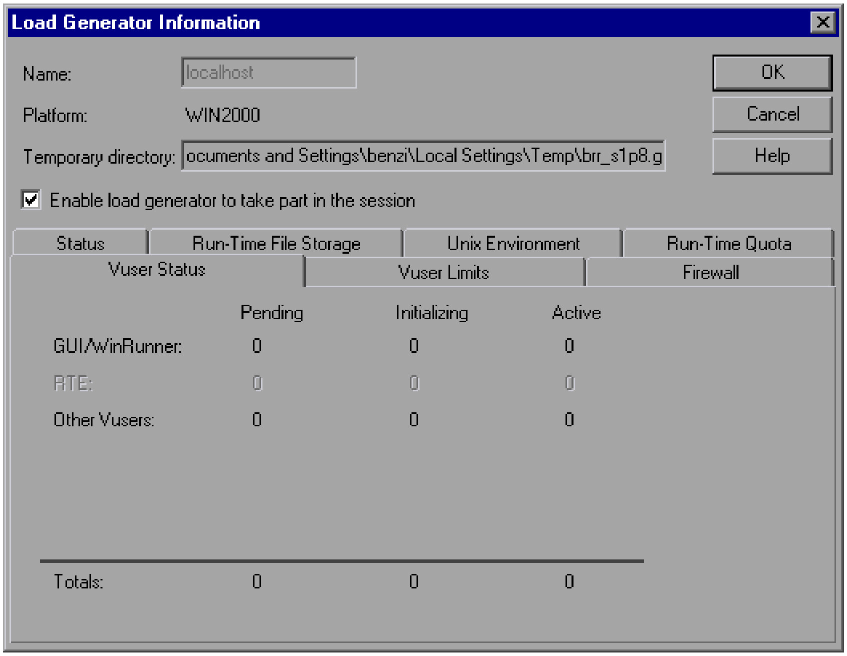
MI Listener: Type the name of the MI listener the load generator is using.

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.



Load Generator Information

Name: localhost

Platform: WIN2000

Temporary directory: ocuments and Settings\benzi\Local Settings\Temp\brr_s1p8.g

☑ Enable load generator to take part in the session

	Pending	Initializing	Active
GUI/WinRunner:	0	0	0
RTE:	0	0	0
Other Vusers:	0	0	0
Totals:	0	0	0

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 493.

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 Mercury Tuning Console 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). Mercury Tuning Console runs the step, adding Vusers, till the step reaches the goal.

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

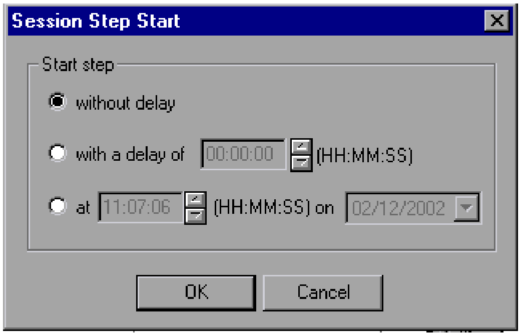
Specifying Execution Time

By default, when you issue an *Execute* command, Mercury Tuning Console immediately starts the session step. You can instruct Mercury Tuning Console 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 Mercury Tuning Console 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.



Start step

● without delay

○ with a delay of 00:00:00 (HH:MM:SS)

○ at 11:07:06 (HH:MM:SS) on 02/12/2002

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 136 and “Understanding the Step Duration Tab” on page 137.

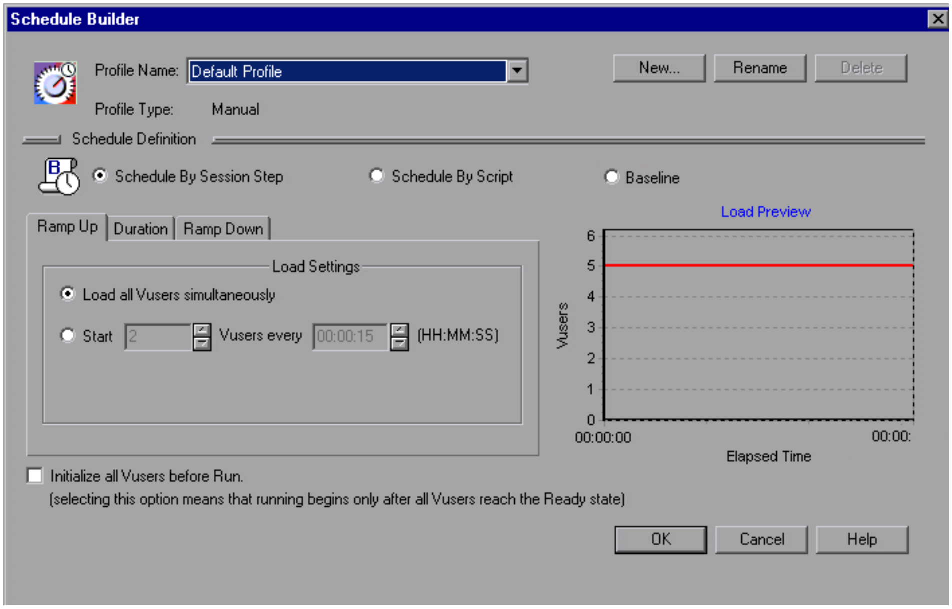
Creating and Selecting a Profile

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, Mercury Tuning Console by default uses a manual profile and assigns it the name *Default Profile*.



Schedule Builder

Profile Name: Default Profile

Profile Type: Manual

Schedule Definition

● Schedule By Session Step ○ Schedule By Script ○ Baseline

Load Settings

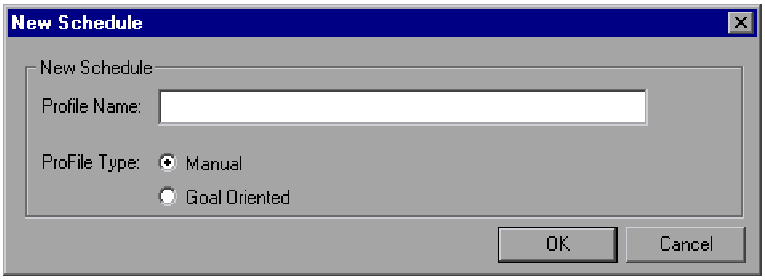
● Load all Vusers simultaneously

○ Start 2 Vusers every 00:00:15 (HH:MM:SS)

□ Initialize all Vusers before Run.
(selecting this option means that running begins only after all Vusers reach the Ready state)

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.



New Schedule

New Schedule

Profile Name:

ProFile Type: ● Manual ○ Goal Oriented

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 134 or “Creating a Goal-Oriented Profile” on page 141).

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.

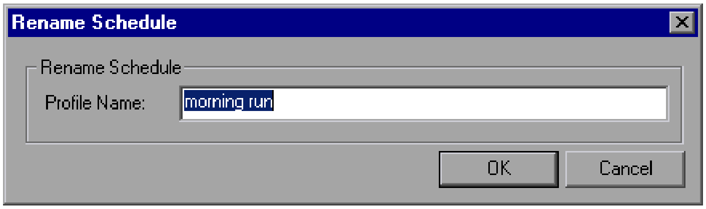
2 Choose the profile you want to modify from the Profile Name list.

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.



Rename Schedule

Rename Schedule

Profile Name: morning run

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. Mercury Tuning Console 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, Mercury Tuning Console 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, Mercury Tuning Console displays radio buttons for choosing the scheduling method:

- Schedule By Session Step
- Schedule By Script
- Baseline

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

When you choose **Schedule by Session Step**, Mercury Tuning Console displays the following tabs:

- Ramp Up tab (see “Understanding the Ramp Up Tab” on page 136).
- Duration tab (see “Understanding the Step Duration Tab” on page 137).
- Ramp Down tab (see “Understanding the Ramp Down Tab” on page 138).

When you choose **Schedule by Script**, you see the Start Time tab (see "Understanding the Start Time Tab" on page 139), 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 141).

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 146).
- Load Behavior Tab (see "Understanding the Load Behavior Tab" on page 147).

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:

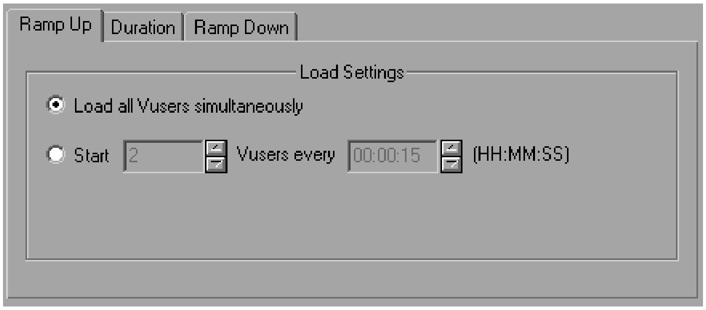
- how the step should be started (ramped up). This allows you to choose between gradually adding Vusers to the running test, or starting all the Vusers simultaneously when the test starts.
- the step duration
- how the step should be stopped (ramped down). This allows you to choose between gradually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



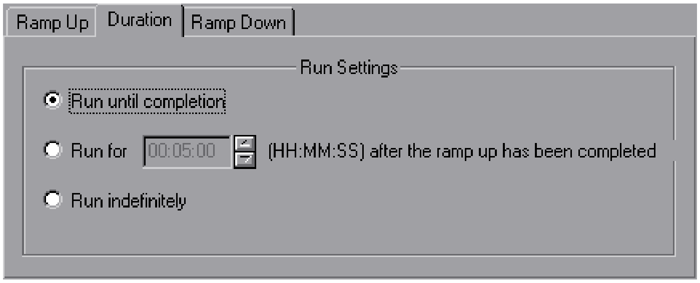
Load Settings

● Load all Vusers simultaneously

○ Start 2 Vusers every 00:00:15 (HH:MM:SS)

For information on the Ramp Up tab, see "Understanding the Ramp Up Tab" on page 136.

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



Run Settings

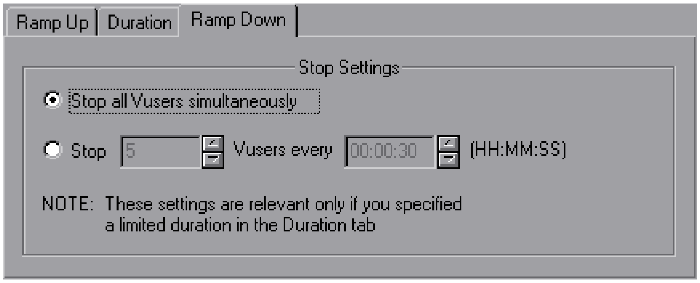
● Run until completion

○ Run for 00:05:00 (HH:MM:SS) after the ramp up has been completed

○ Run indefinitely

For information on the Duration tab, see "Understanding the Step Duration Tab" on page 137.

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



Ramp Up | Duration | Ramp Down

Stop Settings

● Stop all Vusers simultaneously

○ Stop 5 Vusers every 00:00:30 (HH:MM:SS)

NOTE: These settings are relevant only if you specified a limited duration in the Duration tab

For information on the Ramp Down tab, see "Understanding the Ramp Down Tab" on page 138.

To instruct Mercury Tuning Console to initialize Vusers before beginning to load them:

1 Check the **Initialize all Vusers before run** box. Mercury Tuning Console 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 Mercury Tuning Console 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. Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console 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.

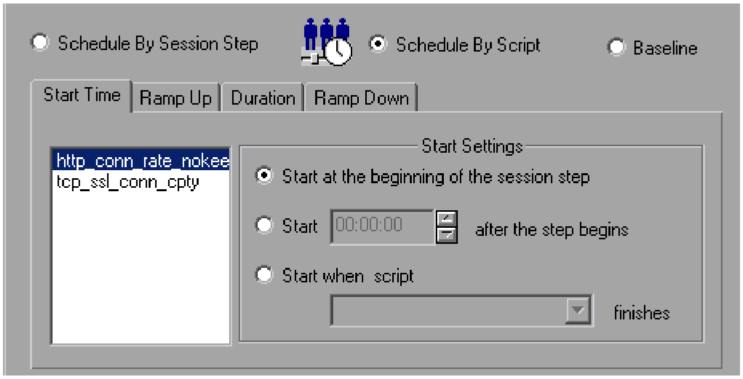
Script Scheduling

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.

Mercury Tuning Console displays a Start Time tab, in addition to the tabs displayed when you schedule by session step (see “Session Step Scheduling” on page 134).



○ Schedule By Session Step ● Schedule By Script ○ Baseline

http_conn_rate_nokee
tcp_ssl_conn_cpty

Start Settings

● Start at the beginning of the session step

○ Start 00:00:00 after the step begins

○ Start when script finishes

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, as described in “Understanding the Start Time Tab” below.

3 Specify the settings in the other tabs (Ramp Up, Duration and Ramp Down) as described in “Scheduling Session Steps,” on page 127.

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. Mercury Tuning Console will run the script when the specified period has passed after the step has begun executing.

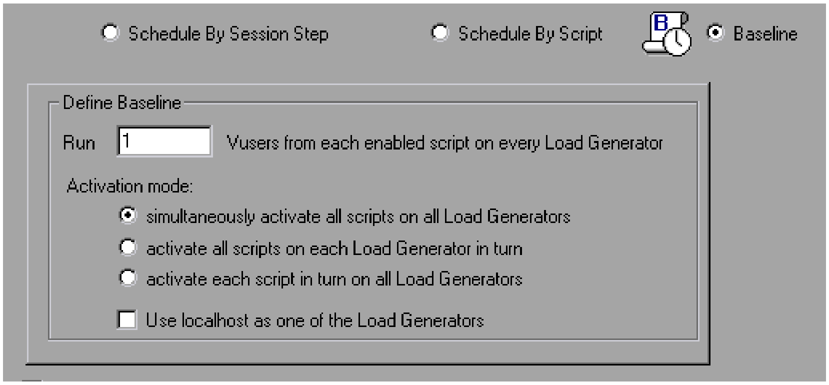
Start when script finishes: Makes the script dependent on another script in the step finishing execution. After clicking the radio button, choose a script from the list box. Mercury Tuning Console will run your script only after the specified script has been run.

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. Mercury Tuning Console displays the Define Baseline pane:



○ Schedule By Session Step ○ Schedule By Script ● Baseline

Define Baseline

Run 1 Vusers from each enabled script on every Load Generator

Activation mode:

● simultaneously activate all scripts on all Load Generators

○ activate all scripts on each Load Generator in turn

○ activate each script in turn on all Load Generators

□ Use localhost as one of the Load Generators

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

3 Specify the activation mode, as described in “Understanding the Baseline Pane” below.

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 Mercury Tuning Console 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, Mercury Tuning Console 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, Mercury Tuning Console 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. Mercury Tuning Console 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. Mercury Tuning Console 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 Mercury Tuning Console should run the step and what it should do if the goal is not reached. **Note:** Times in the scheduler are approximate; although Mercury Tuning Console 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 Mercury Tuning Console to reach your target. Mercury Tuning Console starts each step by ramping up.

  When Mercury Tuning Console 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, Mercury Tuning Console 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.

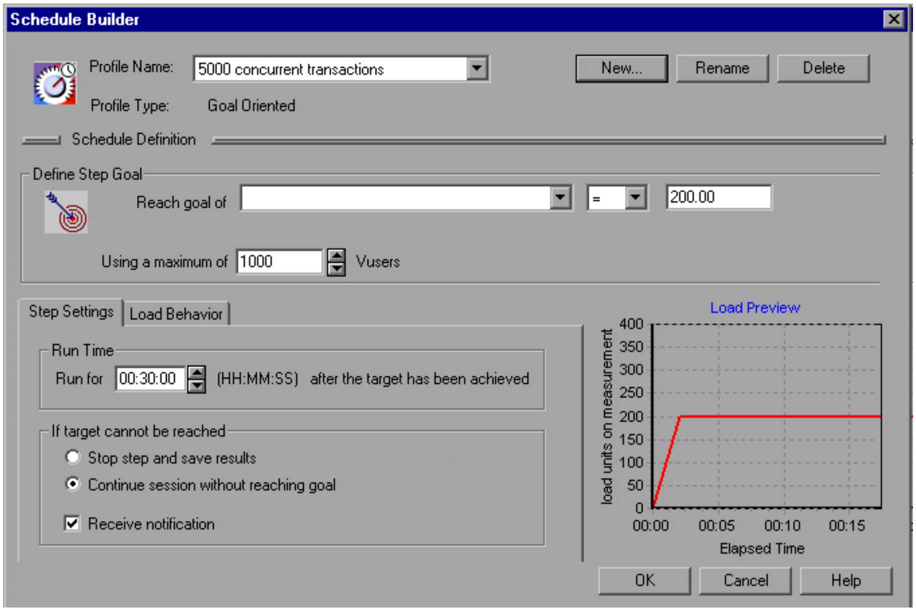
- If you select *Automatic*, the rest of the Vusers needed are run immediately. For example, if Mercury Tuning Console was running one Vuser during the initial two minutes, and it calculates that a total of ten Vusers are needed to reach the goal, it starts another nine Vusers and runs them all for another two minutes. If the goal is not reached by the end of this period, Mercury Tuning Console 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, Mercury Tuning Console 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, Mercury Tuning Console does not consider the goal to have been reached.

- If you select *Reach target after*, Mercury Tuning Console spreads the starting of the Vusers over the time period you specify. For example, if you specify ten minutes and Mercury Tuning Console calculates that it needs five Vusers, one Vuser will be started every two minutes. Note that Mercury Tuning Console 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, Mercury Tuning Console increases the number of Vusers in the batch. Similarly, if the current batch causes a greater change than the previous one, Mercury Tuning Console decreases the number of Vusers in the batch.

  Once the goal has been reached, Mercury Tuning Console 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, Mercury Tuning Console 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:



Schedule Builder

Profile Name: 5000 concurrent transactions

Profile Type: Goal Oriented

Schedule Definition

Define Step Goal

Reach goal of = 200.00

Using a maximum of 1000 Vusers

Run Time

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

If target cannot be reached

○ Stop step and save results

● Continue session without reaching goal

☑ Receive notification

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.

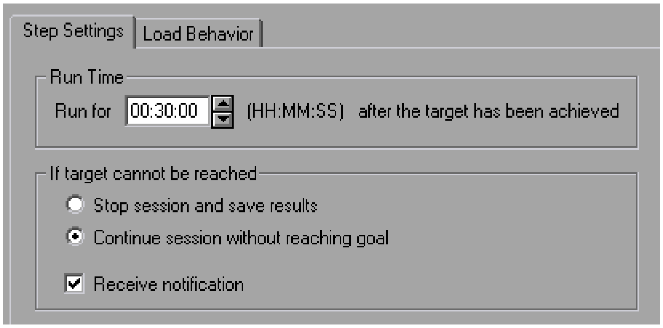
2 Choose an operator (either "=" or ">=")from the middle list box.

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.



Run Time

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

If target cannot be reached

○ Stop session and save results

● Continue session without reaching goal

☑ Receive notification

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 Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console 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.

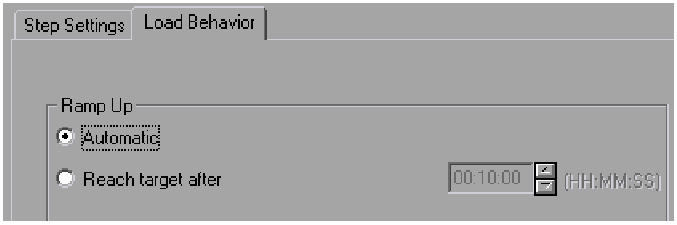
➤ **Continue session without reaching goal:** Instructs the Console to continue running the session, even if the target you defined cannot be reached.

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.



Step Settings | Load Behavior

Ramp Up

● Automatic

○ Reach target after 00:10:00 (HH:MM:SS)

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.

Mercury Tuning Console starts each step by ramping up. Mercury Tuning Console 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, Mercury Tuning Console 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.) Mercury Tuning Console 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 134).

13

Defining Alerts

Before tuning your session, you set up alerts that define what actions Mercury Tuning Console 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

Mercury Tuning Console 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 Mercury Tuning Console 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

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

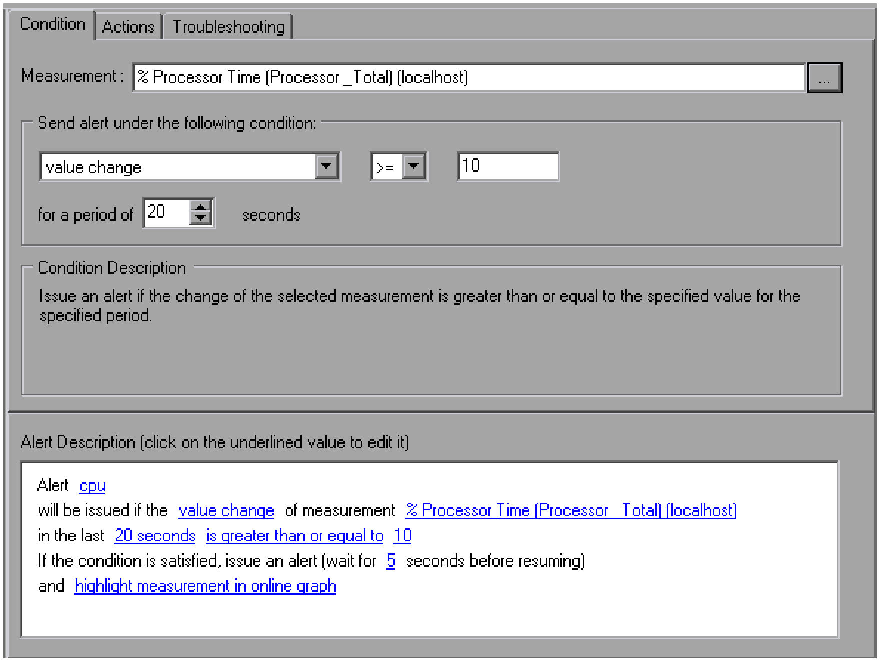
Scheme	Additional fields	Meaning
value	none	Issue an alert if the measurement compares with the specified value in one of the following ways: > >= < <= =
value for a duration of time	for a period of *n* seconds	Issue an alert if the selected measurement compares with the specified value in one of the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



Measurement : % Processor Time (Processor _Total) (localhost)

Send alert under the following condition:

value change >= 10

for a period of 20 seconds

Condition Description

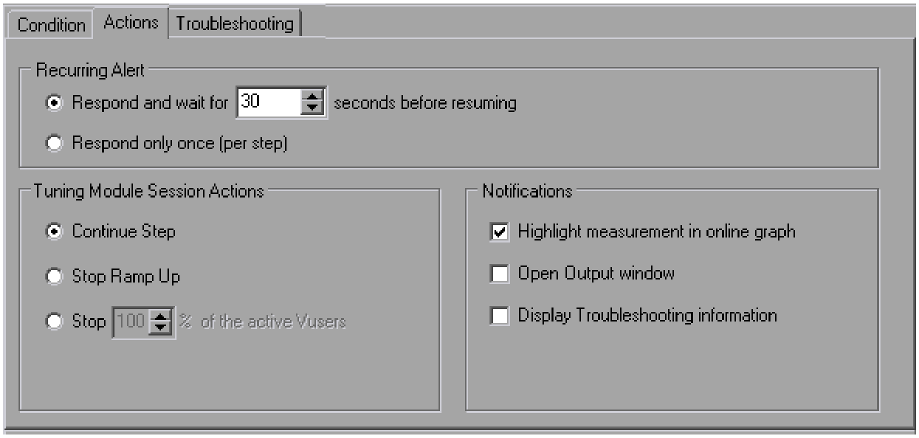
Issue an alert if the change of the selected measurement is greater than or equal to the specified value for the specified period.

Alert Description (click on the underlined value to edit it)

Alert cpu
will be issued if the value change of measurement % Processor Time (Processor _Total) (localhost)
in the last 20 seconds is greater than or equal to 10
If the condition is satisfied, issue an alert (wait for 5 seconds before resuming)
and highlight measurement in online graph

Alert Actions

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



Recurring Alert

● Respond and wait for 30 seconds before resuming

○ Respond only once (per step)

Tuning Module Session Actions

● Continue Step

○ Stop Ramp Up

○ Stop 100 % of the active Vusers

Notifications

☑ Highlight measurement in online graph

□ Open Output window

□ Display Troubleshooting information

Mercury Tuning Console also assigns default settings for the actions that are triggered by an alert. Mercury Tuning Console 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 Mercury Tuning Console 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.

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

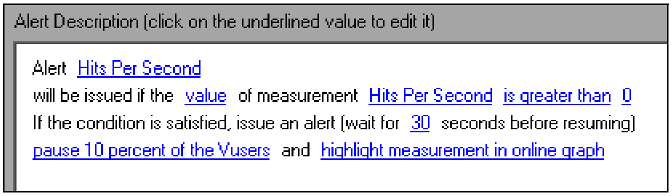
- **Continue Step:** Continues executing the current step, even after an alert trigger.
- **Stop Ramp Up:** Stop adding additional Vusers to the session step.
- **Stop *n* % of the active Vusers:** Stops a percentage of the Vusers when an alert is triggered. Specify a value for *n* between 0 and 100. (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 165.
- **Display Troubleshooting information:** Displays troubleshooting information when an alert trigger occurs. See "Alert Troubleshooting Information" on page 163 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. Mercury Tuning Console places the cursor in the relevant field within the Alerts window.



Alert Description (click on the underlined value to edit it)

Alert Hits Per Second
will be issued if the value of measurement Hits Per Second is greater than 0
If the condition is satisfied, issue an alert (wait for 30 seconds before resuming)
pause 10 percent of the Vusers and highlight measurement in online graph

In some cases, clicking a hyperlink causes Mercury Tuning Console 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. In the Alert Options dialog box, select a trigger option and click **OK**. For more information, see the section "Enabling and Disabling the Alert Mechanism" on page 164.

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.

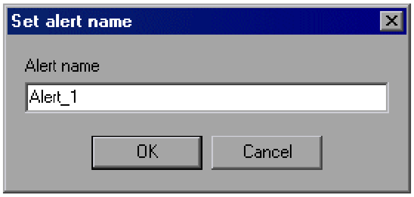
Creating an Alert

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

To create an alert:

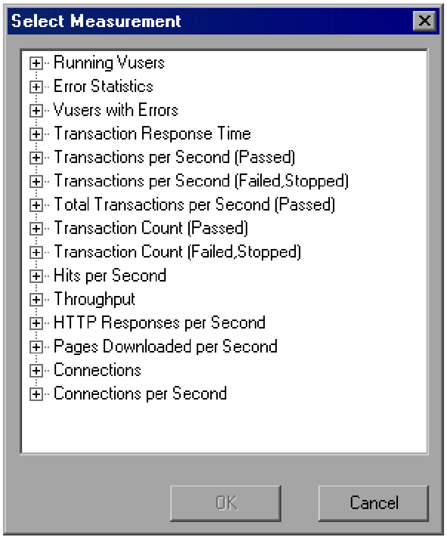
1 Click the **Alerts Definition** button in the Mercury Tuning Console window. The Alerts window opens.

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


Set alert name

Alert name: Alert_1

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


Select Measurement

- Running Vusers
- Error Statistics
- Vusers with Errors
- Transaction Response Time
- Transactions per Second (Passed)
- Transactions per Second (Failed,Stopped)
- Total Transactions per Second (Passed)
- Transaction Count (Passed)
- Transaction Count (Failed,Stopped)
- Hits per Second
- Throughput
- HTTP Responses per Second
- Pages Downloaded per Second
- Connections
- Connections per Second

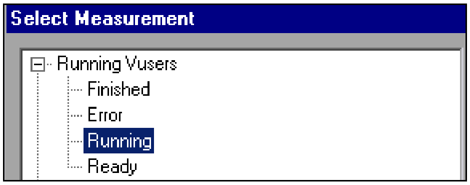
The following categories always appear, regardless of your topology and monitors:

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

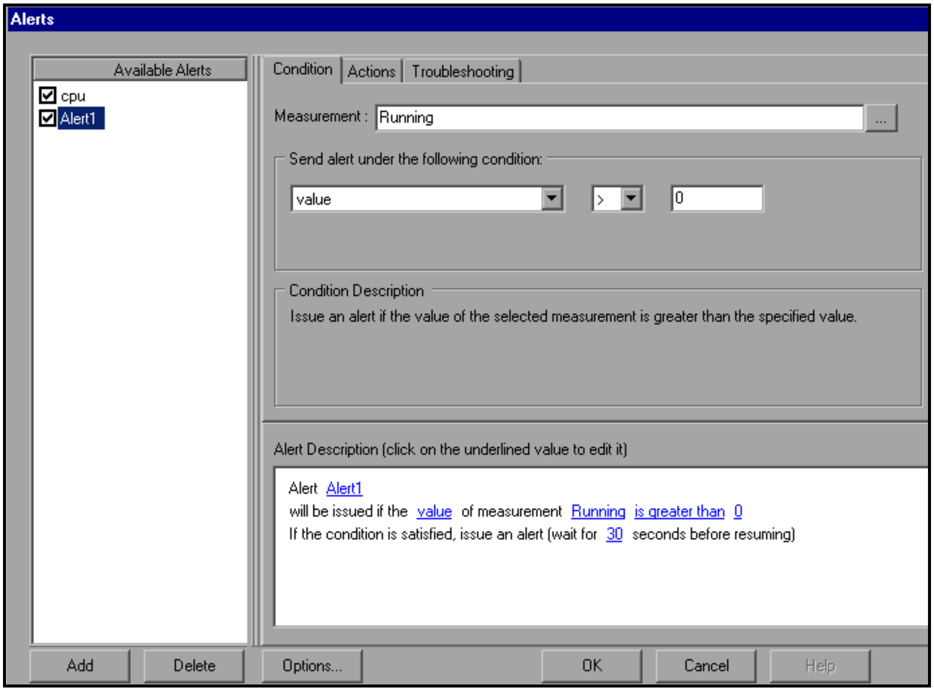
- ➤ Connections per second

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**. Mercury Tuning Console 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.



Measurement : Running

Send alert under the following condition: value > 0

Condition Description

Issue an alert if the value of the selected measurement is greater than the specified value.

Alert Description (click on the underlined value to edit it)

Alert Alert1
will be issued if the value of measurement Running is greater than 0
If the condition is satisfied, issue an alert (wait for 30 seconds before resuming)

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 161 and "Understanding the Actions Tab" on page 155.

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, Mercury Tuning Console 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 Mercury Tuning Console alerts. For more details, see "Enabling and Disabling the Alert Mechanism," on page 164.

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 150.

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 Mercury Tuning Console 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.

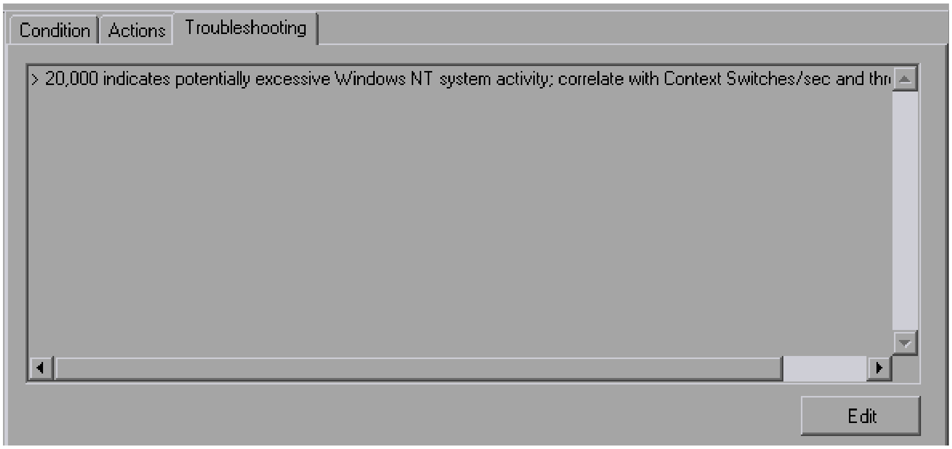
Condition Description: Displays the description of the condition that will trigger an alert.

Note: If you want Mercury Tuning Console 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. Mercury Tuning Console 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.



> 20,000 indicates potentially excessive Windows NT system activity; correlate with Context Switches/sec and thr

To cause Mercury Tuning Console to display the troubleshooting information when the alert is triggered:

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

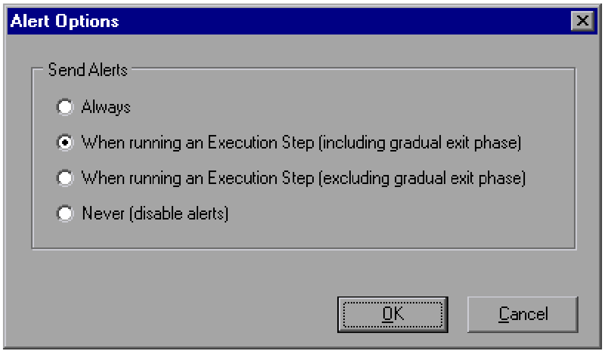
You can add to or change the information that is displayed.

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 Mercury Tuning Console to generate them. In the Alerts Window, click **Options**. The Alert Options dialog box is displayed.



Send Alerts

○ Always

● When running an Execution Step (including gradual exit phase)

○ When running an Execution Step (excluding gradual exit phase)

○ Never (disable alerts)

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:

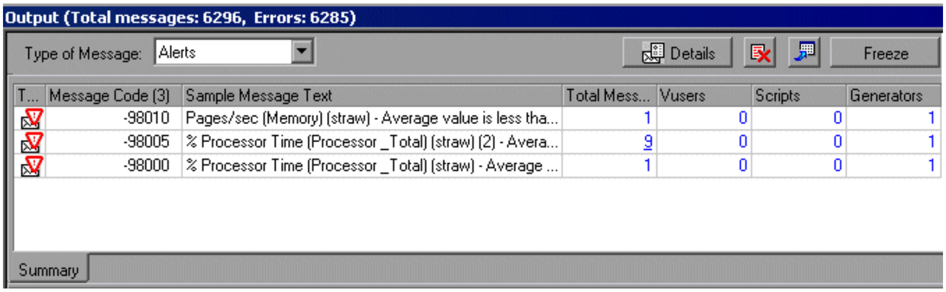
- **Always:** Always issue alerts when the alert condition is met, even when Mercury Tuning Console is not executing a step.
- **When Running an Execution step (including gradual exit phase):** Issue alerts only during test execution, even during the Vuser's gradual exit stage.
- **When Running an Execution 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 (disable alerts):** 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.



Output (Total messages: 6296, Errors: 6285)

Type of Message: Alerts

T...	Message Code (3)	Sample Message Text	Total Mess...	Vusers	Scripts	Generators
	-98010	Pages/sec (Memory) (straw) - Average value is less tha...	1	0	0	1
	-98005	% Processor Time (Processor _Total) (straw) (2) - Avera...	9	0	0	1
	-98000	% Processor Time (Processor _Total) (straw) - Average ...	1	0	0	1

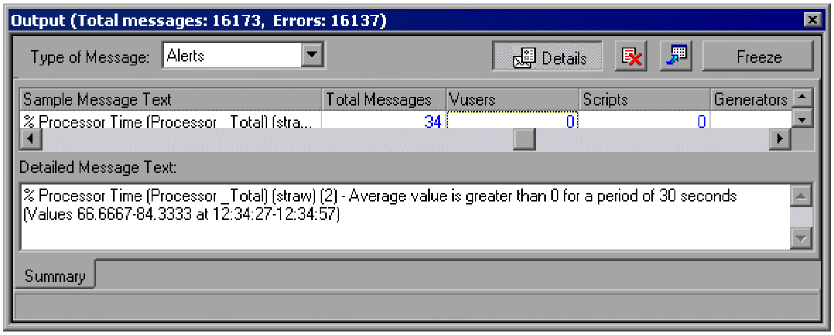
2 If the **Type of Message** field does not show the word “Alerts”, choose it from the drop-down list.

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.



Output (Total messages: 16173, Errors: 16137)

Type of Message: Alerts

Sample Message Text	Total Messages	Vusers	Scripts	Generators
% Processor Time (Processor _Total) (stra...	34	0	0	

Detailed Message Text:

% Processor Time (Processor _Total) (straw) (2) - Average value is greater than 0 for a period of 30 seconds (Values 66.6667-84.3333 at 12:34:27-12:34:57)

For more details on using the Output window, see “Viewing the Output Window” on page 230.

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. You can also enable the Web page breakdown feature.

This chapter describes:

- Configuring Session Run-Time Settings
- Setting Timeout Intervals
- Setting the Run-Time File Location
- Specifying Path Translation
- Enabling Web Page Breakdown

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, Mercury Tuning Console 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 *Mercury Virtual User Generator User's Guide*.

For information on setting the options for online monitors, see Chapter 27, "Online Monitoring."

In order for the Analysis to generate Web Page Breakdown graphs, you need to enable the Web page breakdown feature before running your scenario or session step. See "Enabling Web Page Breakdown," on page 178.

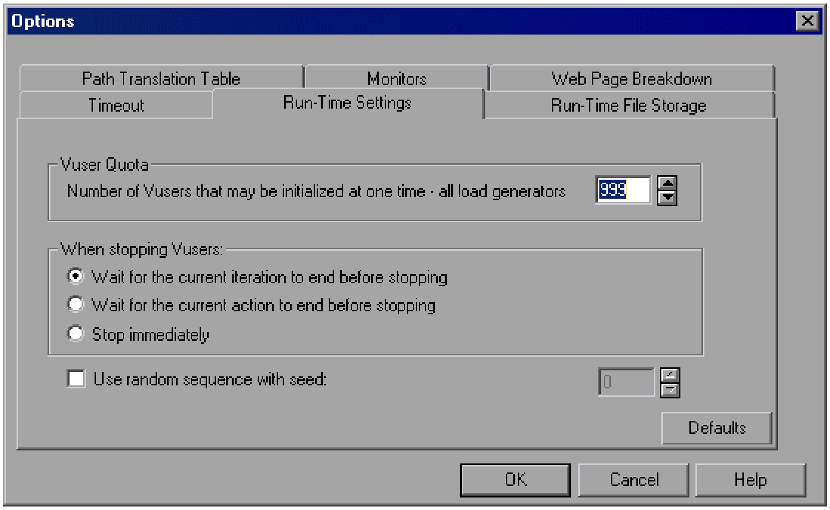
The Mercury Tuning Console Expert mode allows you to configure additional settings for the Mercury Tuning Console agent and other Mercury Tuning Console components. For more information, see Appendix C, "Working in Expert Mode."

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.



Options

Vuser Quota

Number of Vusers that may be initialized at one time - all load generators 999

When stopping Vusers:

● Wait for the current iteration to end before stopping

○ Wait for the current action to end before stopping

○ Stop immediately

□ Use random sequence with seed: 0

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

3 Select the way in which you want Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console to stop running the Vuser(s) immediately. The Vuser(s) move to the EXITING status and immediately stop executing.

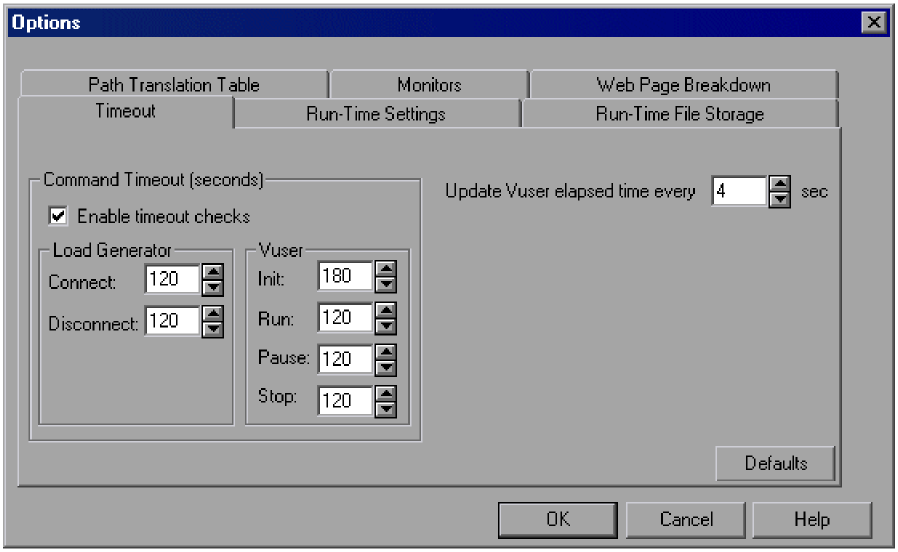
Use random sequence with seed: Allows Mercury Tuning Console to use a seed number for random sequencing. Each seed value represents one sequence of random values used for test execution. Whenever you use this seed value, the same sequence of values is assigned to the Vusers in the step. This setting 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.



Command Timeout (seconds)

☑ Enable timeout checks

Load Generator

Connect: 120

Disconnect: 120

Vuser

Init: 180

Run: 120

Pause: 120

Stop: 120

Update Vuser elapsed time every 4 sec

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 Mercury Tuning Console updates the Elapsed time, in the **Update Vuser elapsed time every** box.

Understanding the Options - Timeout Tab

Mercury Tuning Console enables you to set the timeout interval for commands and Vuser elapsed time.

The command timeouts are the maximum time limits for various Mercury Tuning Console 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 Mercury Tuning Console 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, Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console 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: Mercury Tuning Console 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. Mercury Tuning Console 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 Mercury Tuning Console 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, Mercury Tuning Console 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.

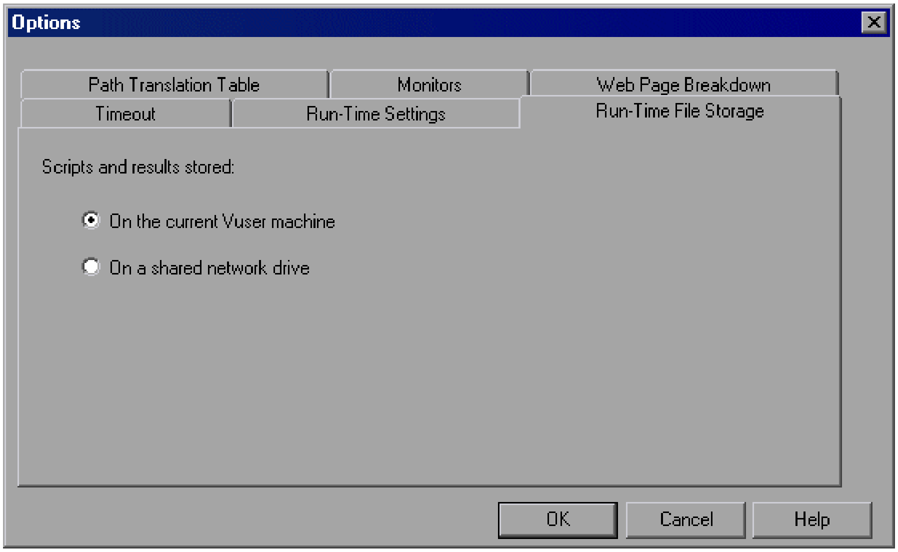
Note: The run-time file storage settings that are described in this apply to all the load generators in a session. You can change the settings for individual load generator machines as described in "Configuring Load Generators" on page 109.

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 18, "Running an Execution Step." After collating the results, the temporary run-time directory is deleted.

To specify where Mercury Tuning Console stores run-time files:

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



Scripts and results stored:

● On the current Vuser machine

○ On a shared network drive

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 Mercury Tuning Console 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 Mercury Tuning Console 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

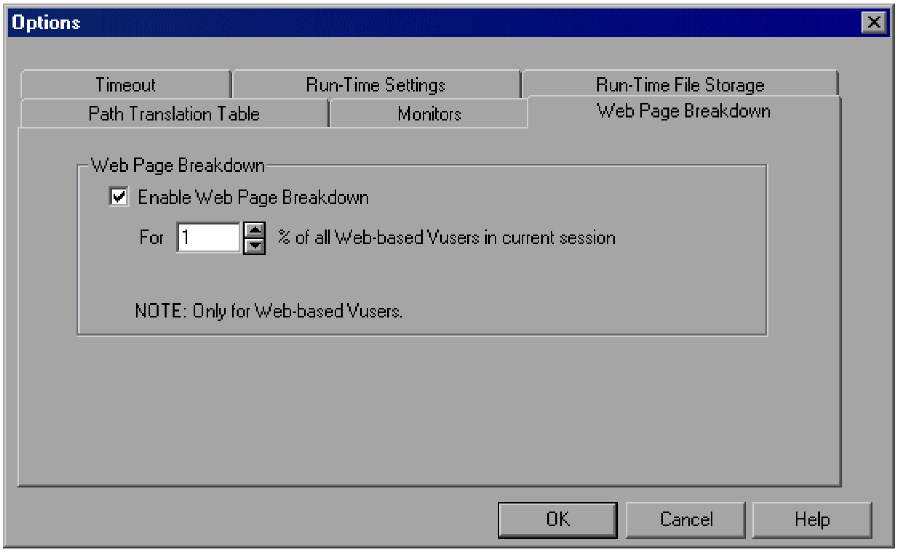
If you specified a shared network drive for run-time file storage, (see “Setting the Run-Time File Location” on page 174), you may need to perform *path translation*. Path translation is a mechanism used by Mercury Tuning Console to convert a remote path names. A typical session may contain several load generator machines that map the shared network drive differently. For more information, see Appendix D, “Performing Path Translation.”

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 Tuning Console before running your session step.

Enabling Web Page Breakdown in the Tuning Console

From the Console menu, choose **Tools** > **Options** and select the **Web Page Breakdown** tab.



Web Page Breakdown

☑ Enable Web Page Breakdown

For 1 % of all Web-based Vusers in current session

NOTE: Only for Web-based Vusers.

Enable Web Page Breakdown: Enables you to view the Web Page Breakdown graphs in the Analysis. This feature is only applicable to Web-based Vusers.

- **For X % of all Web-based Vusers in current session:** Specify the percentage of Web Vusers for which you want transaction breakdown to be performed.

For more information about Web Page Breakdown graphs, see the *LoadRunner Analysis User's Guide*.

15

Using Rendezvous Points

Mercury Tuning Console 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 Mercury Tuning Console 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 183.

You define rendezvous points in the Vuser script. For information about inserting rendezvous points into Vuser scripts, refer to the *Mercury Virtual User Generator User’s 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, Mercury Tuning Console 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 183.

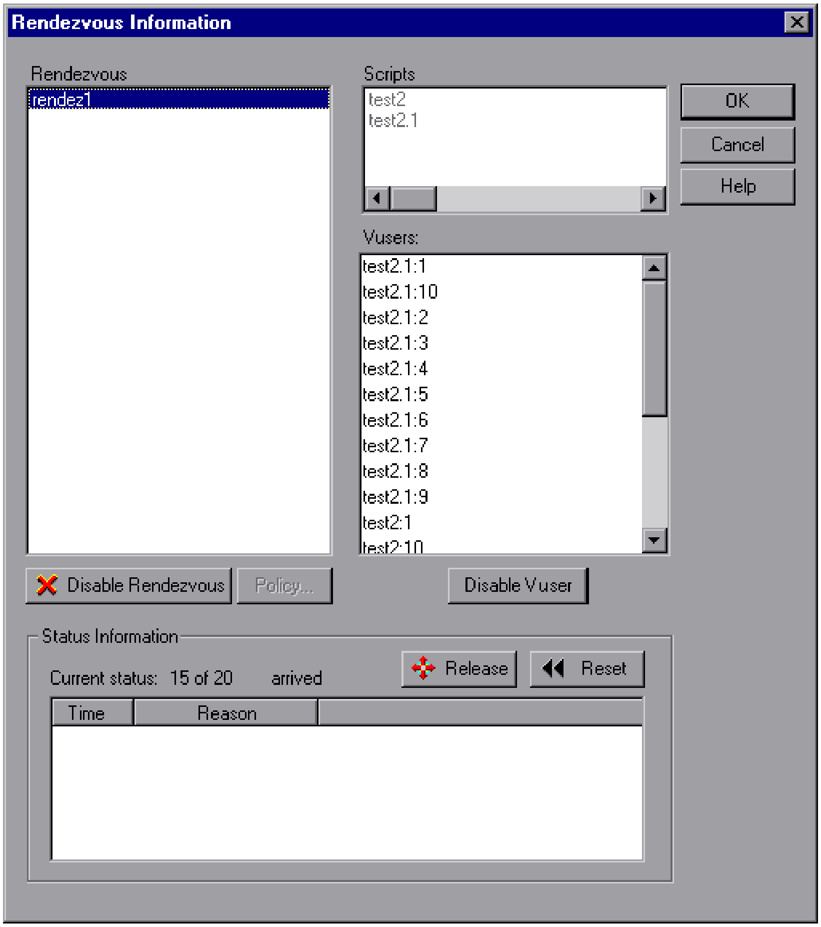
5 Run the session step.

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.



Rendezvous
rendez1

Scripts
test2
test2.1

Vusers:
test2.1:1
test2.1:10
test2.1:2
test2.1:3
test2.1:4
test2.1:5
test2.1:6
test2.1:7
test2.1:8
test2.1:9
test2:1
test2:10

Status Information

Current status: 15 of 20 arrived

Time	Reason

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:

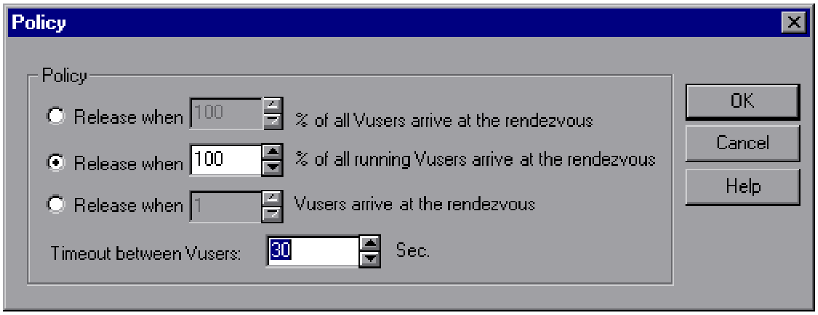
release policy sets how many Vusers will be released from a rendezvous at a time.

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.



Policy

Policy

○ Release when 100 % of all Vusers arrive at the rendezvous

● Release when 100 % of all running Vusers arrive at the rendezvous

○ Release when 1 Vusers arrive at the rendezvous

Timeout between Vusers: 30 Sec.

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.

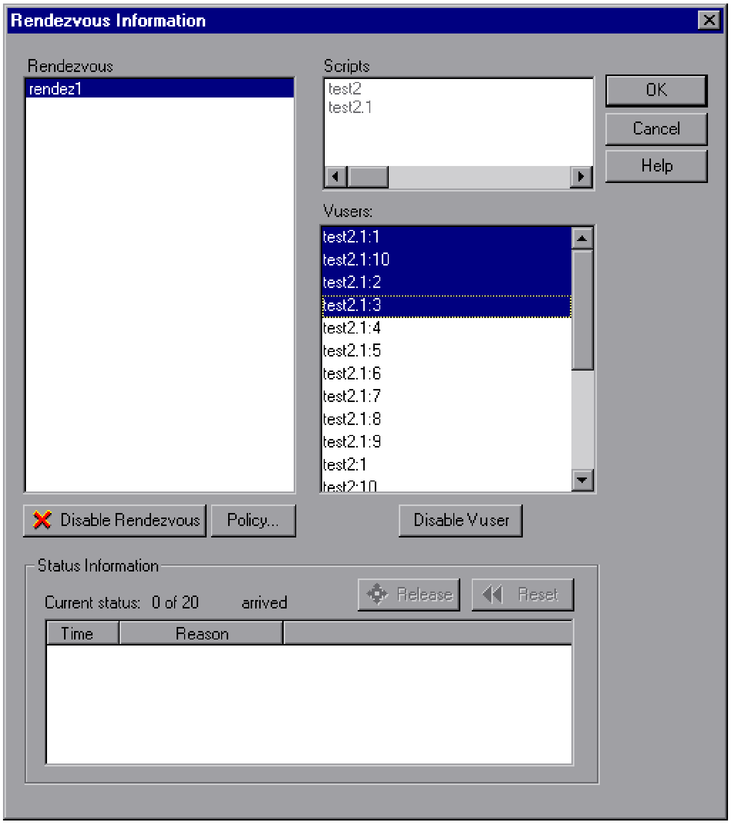
Disabling and Enabling Vusers at Rendezvous Points

In addition to disabling a rendezvous point for all Vusers in a session step, Mercury Tuning Console 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.



Rendezvous: rendez1

Scripts: test2, test2.1

Vusers: test2.1:1, test2.1:10, test2.1:2, test2.1:3, test2.1:4, test2.1:5, test2.1:6, test2.1:7, test2.1:8, test2.1:9, test2:1, test2:10

Current status: 0 of 20 arrived

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:

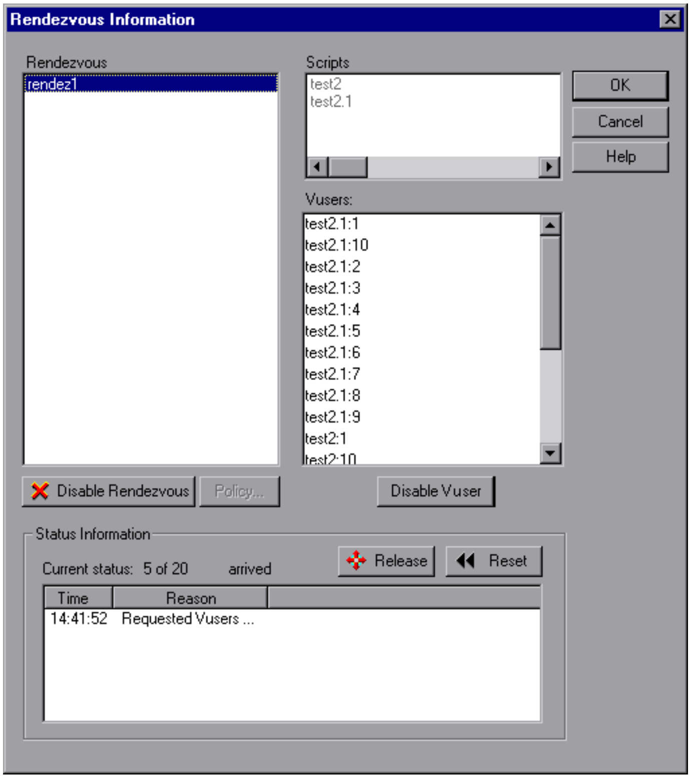
Time: The time at which the Vusers at the rendezvous point were released.

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.



Rendezvous: rendez1

Scripts: test2, test2.1

Vusers:

test2.1:1
test2.1:10
test2.1:2
test2.1:3
test2.1:4
test2.1:5
test2.1:6
test2.1:7
test2.1:8
test2.1:9
test2:1
test2:10

Status Information

Current status: 5 of 20 arrived

Time	Reason
14:41:52	Requested Vusers ...

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, Mercury Tuning Console 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 Mercury Tuning Console session step.

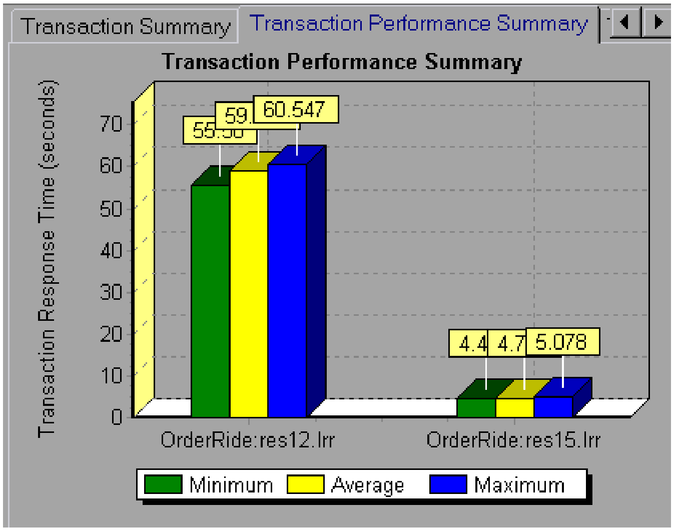
For information on one-time configuration settings such as timeout, output, and quotas, see Chapter 14, "Configuring Session Steps."

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 Mercury Tuning Console 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.”

Mercury Tuning Console allows you to give descriptive names to each result set. This is especially useful for cross results analysis, in which Mercury Tuning Console 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*.

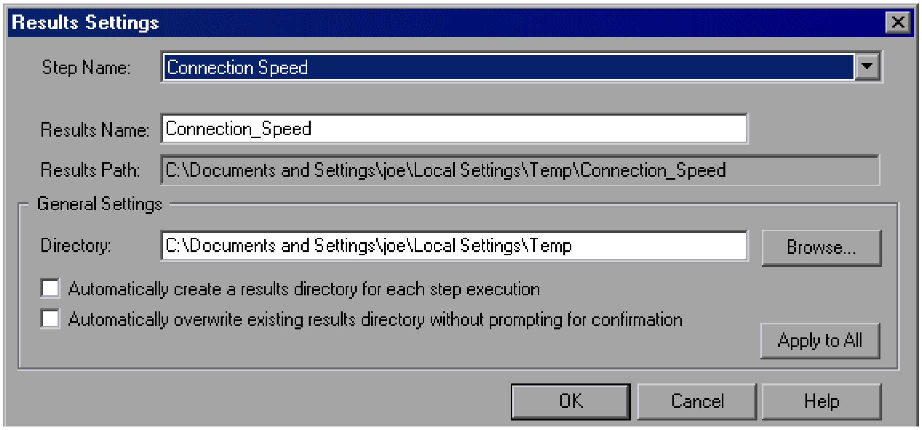


For more details on cross result graphs, refer to the *LoadRunner Analysis User’s Guide*.

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.



Results Settings

Step Name: Connection Speed

Results Name: Connection_Speed

Results Path: C:\Documents and Settings\joe\Local Settings\Temp\Connection_Speed

General Settings

Directory: C:\Documents and Settings\joe\Local Settings\Temp

□ Automatically create a results directory for each step execution

□ Automatically overwrite existing results directory without prompting for confirmation

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, Mercury Tuning Console uses the step name as a basis for the results name.

For example, if the step name is "Connection Speed", Mercury Tuning Console 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 Mercury Tuning Console to create a unique results directory for each session step execution. By default, the result names are *res1*, *res2*, *res3*, etc.

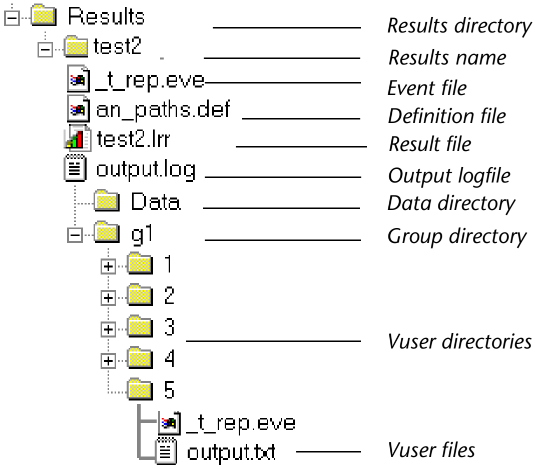
Automatically overwrite existing results directory without prompting for confirmation: Instructs Mercury Tuning Console to automatically overwrite previous result sets, without prompting the user.

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. Mercury Tuning Console 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, Mercury Tuning Console also gathers data from each Vuser and stores it in an event file *_t_rep.eve* and an output file *output.txt.* Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console Analysis, refer to the *LoadRunner 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 192.

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 167 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 494.

The log and result directories are only deleted from a load generator once Mercury Tuning Console 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**. Mercury Tuning Console attempts to collate the results again.

Before generating the analysis data, Mercury Tuning Console 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.

17

Using Scripts Stored in Quality Center

Mercury Tuning Console's integration with Quality Center lets you use Vuser scripts stored in Quality Center.

This chapter describes:

- Connecting to and Disconnecting from Quality Center
- Adding Vuser Scripts from a Quality Center Project

About Using Scripts Stored in Quality Center

Before Mercury Tuning Console can access a Quality Center project, you must connect it to the Web server on which Quality Center is installed. You can connect to either a local or a remote Web server.

For more information on working with Quality Center, refer to the *Quality Center User's Guide*.

Connecting to and Disconnecting from Quality Center

You can connect or disconnect Mercury Tuning Console from a Quality Center project at any time during the tuning process.

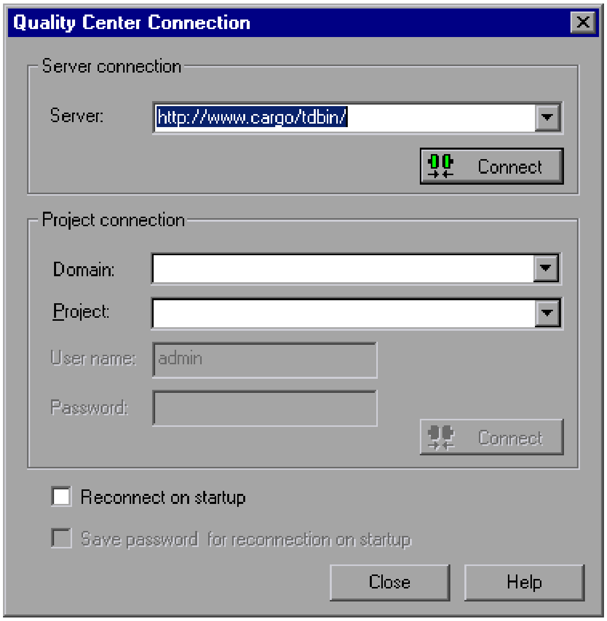
Connecting Mercury Tuning Console to Quality Center

The connection process has two stages. First, you connect Mercury Tuning Console to a local or remote Quality Center Web server. This server handles the connections between Mercury Tuning Console and the Quality Center project.

Next, you choose the project you want Mercury Tuning Console to access. Note that Quality Center projects are password protected, so you need to provide a user name and a password.

To connect Mercury Tuning Console to Quality Center:

1 Choose **Tools** > **Quality Center Connection**. The Quality Center Connection dialog box opens.



Quality Center Connection

Server connection

Server: http://www.cargo/tdbin/

Project connection

Domain:

Project:

User name: admin

Password:

□ Reconnect on startup

□ Save password for reconnection on startup

2 In the Server field, type or choose the URL address of the Web server on which Quality Center is installed, and click the **Connect** button in the Server Connection section.

3 Enter the relevant information in the Project Connection section and click the **Connect** button in that section.

4 Click **Close** to close the Quality Center Connection dialog box.

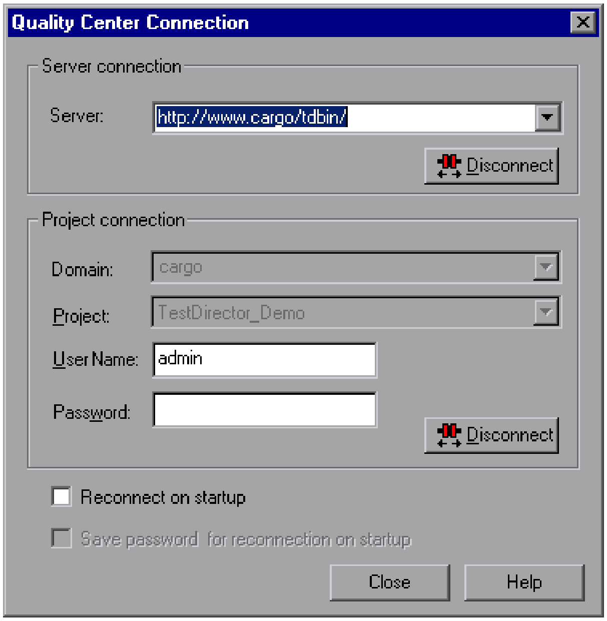
The status bar indicates that Mercury Tuning Console is currently connected to a Quality Center project.

Disconnecting Mercury Tuning Console from Quality Center

You can disconnect Mercury Tuning Console from a selected Quality Center project and Web server.

To disconnect Mercury Tuning Console from Quality Center:

1 Choose **Tools** > **Quality Center Connection**. The Quality Center Connection dialog box opens.



Quality Center Connection

Server connection

Server: http://www.cargo/tdbin/

Disconnect

Project connection

Domain: cargo

Project: TestDirector_Demo

User Name: admin

Password:

Disconnect

□ Reconnect on startup

□ Save password for reconnection on startup

2 To disconnect Mercury Tuning Console from the selected project, click **Disconnect** in the Project Connection section.

3 To disconnect Mercury Tuning Console from the selected server, click **Disconnect** in the Server Connection section.

4 Click **Close** to close the Quality Center Connection dialog box.

Understanding the Quality Center Connection Dialog Box

You use the Quality Center Connection dialog box to open a connection to a Quality Center project.

Server Connection section: Used for opening a connection to the server that hosts the Quality Center project.

- **Server:** The name of the server. This can be a Web server accessible via a Local Area Network (LAN) or a Wide Area Network (WAN).
- **Connect:** Click to connect to the specified server. Once the connection to the server is established, the server's name is displayed in read-only format.

Project Connection section: Used for specifying connection parameters and connecting to the Quality Center project.

- **Domain:** Project's domain name.
- **Project:** Name of the project to which you want to connect. The list includes all the projects that are registered with the selected server. Once the connection to the selected project is established, the project's name is displayed in read-only format.
- **User Name:** Your user name.
- **Password:** Your user password.
- **Connect:** Click to connect to the selected project.

Reconnect on startup: When you check this box, Mercury Tuning Console automatically opens the connection to the Quality Center server and the specified project when you subsequently start the Console. Checking this box also enables the **Save password for reconnection on startup** check box.

Save password for reconnection on startup: Check this box to save the specified password for subsequent connections, thereby automating the login process. If you do not choose this option, you will be prompted to enter the password when Mercury Tuning Console connects to Quality Center on startup.

Adding Vuser Scripts from a Quality Center Project

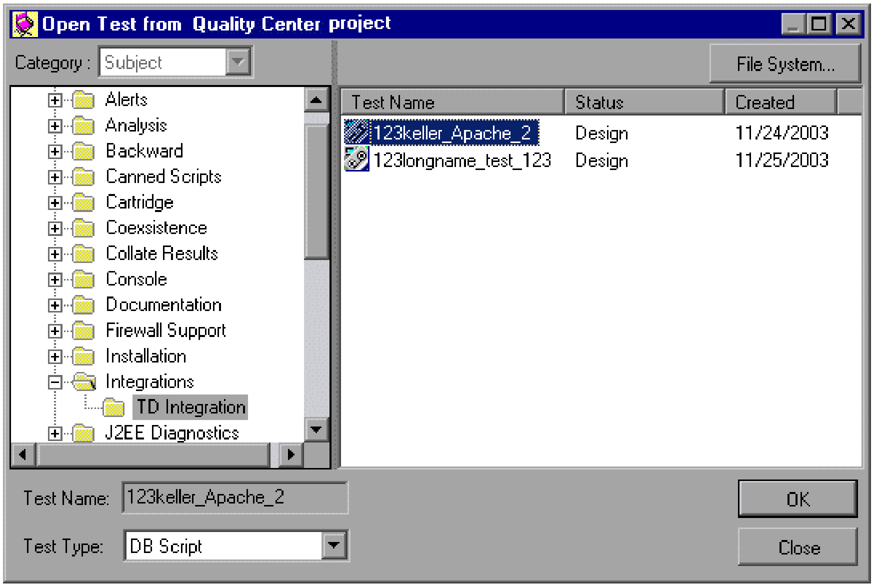
You can add Vuser scripts from a Quality Center project to an Execution step.

To add a Vuser script to an Execution step:

1 Connect to the Quality Center server (see “Connecting Mercury Tuning Console to Quality Center,” on page 199).

2 Create a new Execution step or select an existing one, right-click the step in the session tree, and choose **Add Script**. The Add Script dialog box opens.

3 Click **Add**, and then click **Browse** > **For All Scripts**. The Open Test from Quality Center Project dialog box opens and displays the test plan tree.



Category : Subject

Test Name	Status	Created
123keller_Apache_2	Design	11/24/2003
123longname_test_123	Design	11/25/2003

Test Name: 123keller_Apache_2

Test Type: DB Script

4 Select the script and click **OK**. The Script Path field displays [TD], the full subject path, and the script name. For example:

[TD]\Subject\System\test_td

5 Click **OK**. The script is displayed in the Custom Scripts pane.

6 Click **OK** to close the Add Script dialog box. The script is displayed in the session tree.

Part III

Executing a Tuning Session

18

Running an Execution Step

When you tune a session, Mercury Tuning Console 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, Mercury Tuning Console:

- 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 Mercury Tuning Console 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 169.

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 214.

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 72), depending on whether or not the current step was successful, Mercury Tuning Console 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

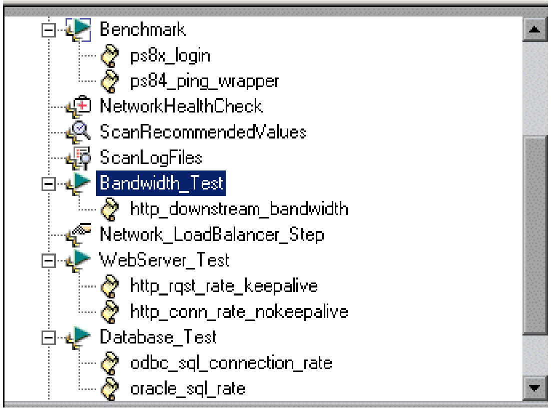
You can run all the Vusers specified by the session step settings, or you can select the number of Vusers that you want to run. Note that when you run your Execution step, Mercury Tuning Console runs the Vusers as soon as they reach the READY state.

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 216.

To run an Execution step:

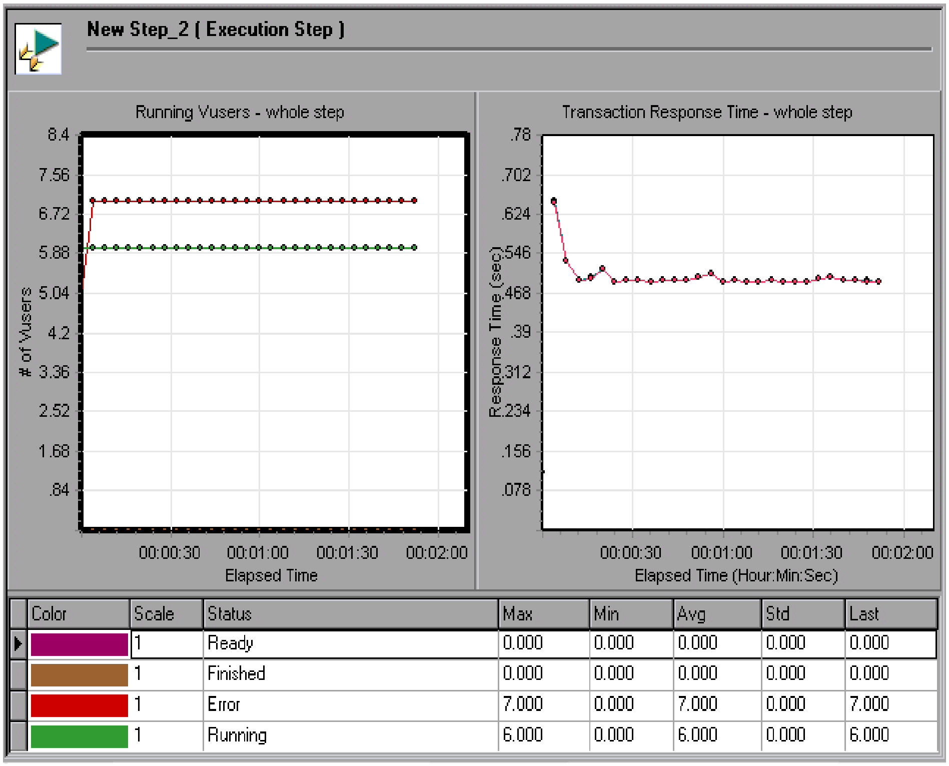
1 Open an existing session or create a new one.

2 In the Session tree, select the step that you want to execute.



```
Benchmark
    ps8x_login
    ps84_ping_wrapper
NetworkHealthCheck
ScanRecommendedValues
ScanLogFiles
Bandwidth_Test
    http_downstream_bandwidth
Network_LoadBalancer_Step
WebServer_Test
    http_rqst_rate_keepalive
    http_conn_rate_nokeepalive
Database_Test
    odbc_sql_connection_rate
    oracle_sql_rate
```

The default monitors are displayed.



	Color	Scale	Status	Max	Min	Avg	Std	Last
▶	🟪	1	Ready	0.000	0.000	0.000	0.000	0.000
	🟫	1	Finished	0.000	0.000	0.000	0.000	0.000
	🟥	1	Error	7.000	0.000	7.000	0.000	7.000
	🟩	1	Running	6.000	0.000	6.000	0.000	6.000

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** button in the Session tab's tool bar. 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 214.

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**).

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 32.

Add/Remove Graphs: Displays the Select Online Graphs dialog box. See “Opening Online Monitor Graphs in the Tuning Console” on page 357.

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:

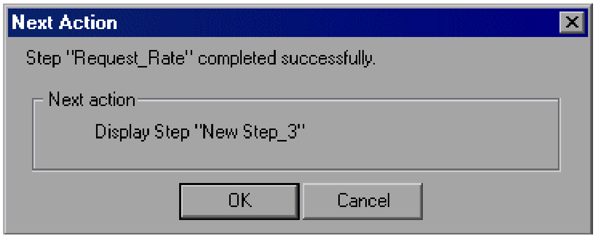
- Show only the selected measurement
- Show/Hide the measurement
- Configure the measurement
- View the measurement's description
- Open the Alerts Definition page for the measurement

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 218.

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 72).

If you specified the subsequent step or steps, Mercury Tuning Console displays the Next Action dialog box when the current step completes execution.



Next Action

Step "Request_Rate" completed successfully.

Next action

Display Step "New Step_3"

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.

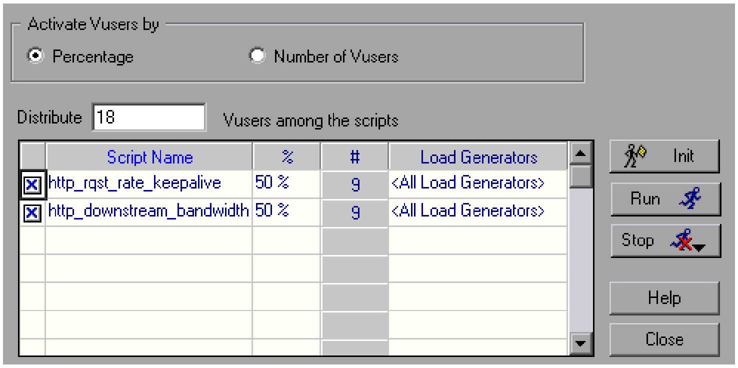
Click **Cancel** to leave the focus on the step that completed execution.

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.



Activate Vusers by

● Percentage ○ Number of Vusers

Distribute 18 Vusers among the scripts

	Script Name	%	#	Load Generators
☒	http_rqst_rate_keepalive	50 %	9	<All Load Generators>
☒	http_downstream_bandwidth	50 %	9	<All Load Generators>

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.

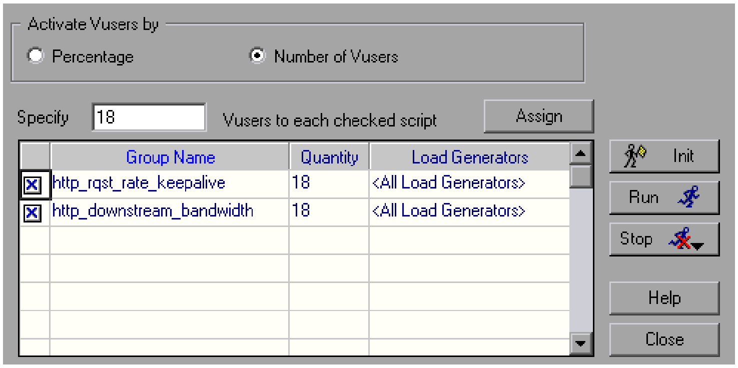
When a step includes more than one script, you can either specify what percentage of the total number of Vusers should execute each script, or specify the number of Vusers. You can start or stop a specified number of Vusers or a percentage of the total.

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. Mercury Tuning Console 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. Mercury Tuning Console automatically adjusts the percentages of the other scripts.

Number of Vusers: Clicking this radio button displays the **Specify ... Vusers to each checked script** box.



Activate Vusers by

○ Percentage ● Number of Vusers

Specify 18 Vusers to each checked script

	Group Name	Quantity	Load Generators
☒	http_rqst_rate_keepalive	18	<All Load Generators>
☒	http_downstream_bandwidth	18	<All Load Generators>

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

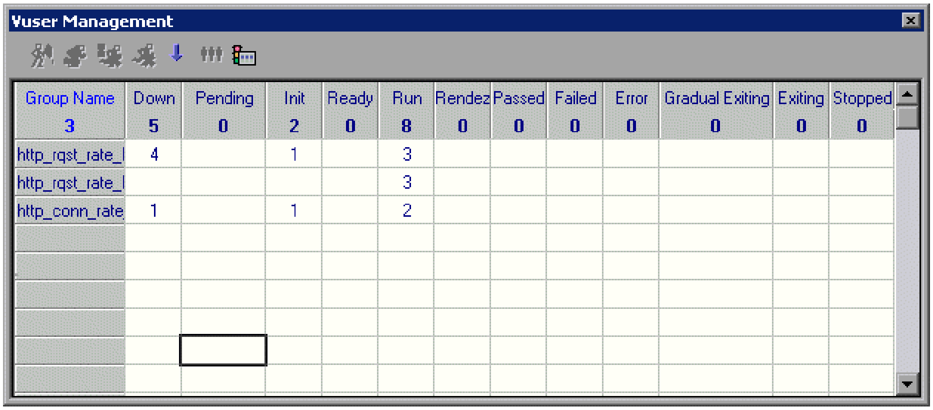
Mercury Tuning Console allows you to see different views of the Vusers running in a step.

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 Mercury Tuning Console 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.)



Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
3	5	0	2	0	8	0	0	0	0	0	0	0
http_rqst_rate_l	4		1		3							
http_rqst_rate_l					3							
http_conn_rate	1		1		2							

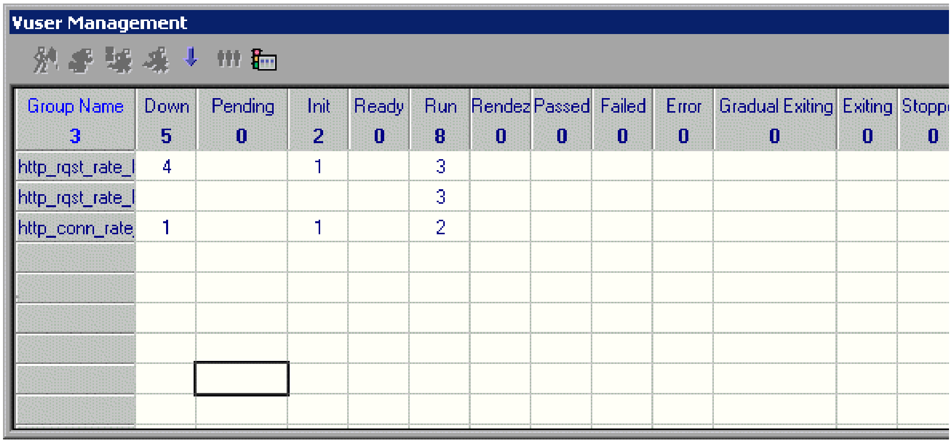
Each row of the table lists a group of Vusers in the session step. The columns show how many of the group's Vusers have a particular status. You can do the following:

- 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.

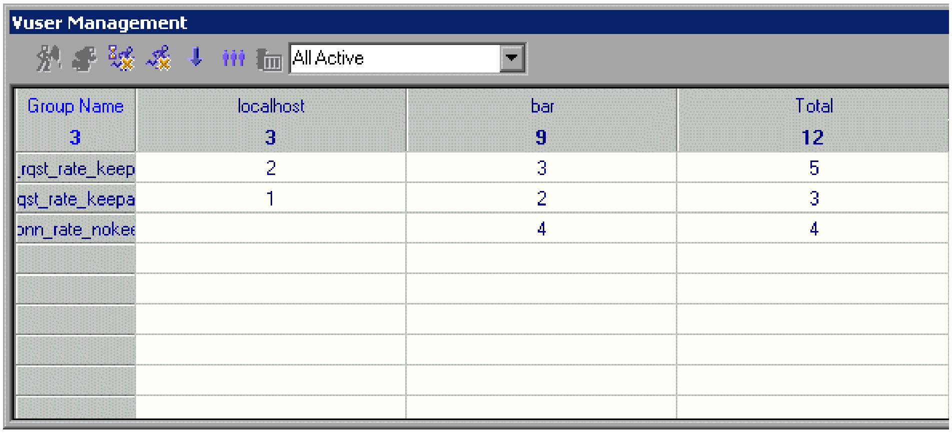
Show Vuser Status By Group: Click this button to view the Vusers in each group, broken down by status (for example, Down, Pending, and so on):



Vuser Management

Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopp
3	5	0	2	0	8	0	0	0	0	0	0	0
http_rqst_rate_l	4		1		3							
http_rqst_rate_l					3							
http_conn_rate	1		1		2							

Show Groups by Load Generator: Click this button to view the number of Vusers that each group is running on each load generator.



Vuser Management

All Active

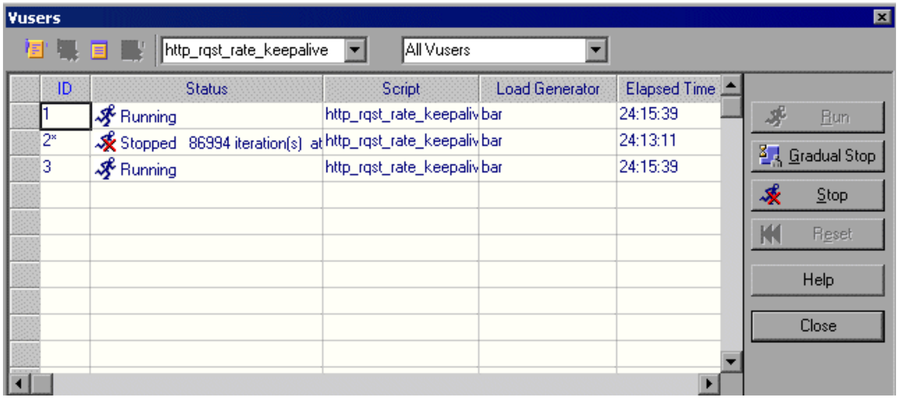
Group Name	localhost	bar	Total
3	3	9	12
rqst_rate_keep	2	3	5
qst_rate_keepa	1	2	3
onn_rate_nokee		4	4

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.



http_rqst_rate_keepalive

All Vusers

ID	Status	Script	Load Generator	Elapsed Time
1	Running	http_rqst_rate_keepaliv	bar	24:15:39
2*	Stopped 86994 iteration(s) at	http_rqst_rate_keepaliv	bar	24:13:11
3	Running	http_rqst_rate_keepaliv	bar	24:15:39

Note: To choose a different Vuser group, select it from the left list box.

Understanding the Vusers in Group Dialog Box

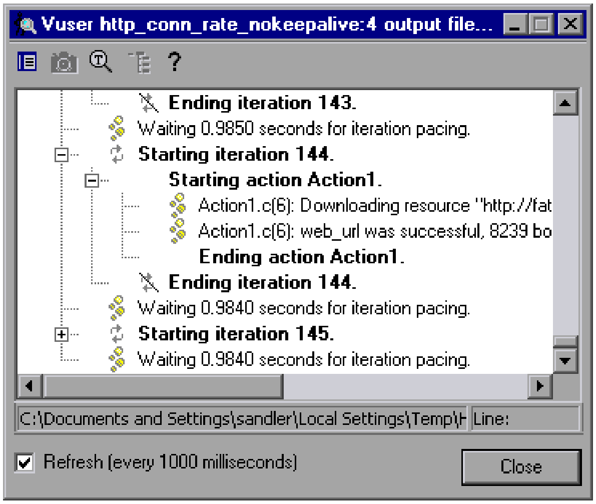
The Vusers in Group dialog box lists a group's Vusers, and shows you the following information about each: ID, Status, Script, Load Generator, and Elapsed Time (since the beginning of the session). It also allows you to control each Vuser's activity.

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 223.

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.



```
        Ending iteration 143.
      Waiting 0.9850 seconds for iteration pacing.
      Starting iteration 144.
        Starting action Action1.
          Action1.c(6): Downloading resource "http://fat
          Action1.c(6): web_url was successful, 8239 bo
          Ending action Action1.
        Ending iteration 144.
      Waiting 0.9840 seconds for iteration pacing.
      Starting iteration 145.
      Waiting 0.9840 seconds for iteration pacing.
```

C:\Documents and Settings\sandler\Local Settings\Temp\F Line:

☑ Refresh (every 1000 milliseconds)

The log information is refreshed, by default, every 1000 milliseconds.

For more information on the script log, see page 236.

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.
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 Mercury Tuning Console.
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, Mercury Tuning Console 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 170.
- **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 tool bars and views.

19

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

Mercury Tuning Console 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

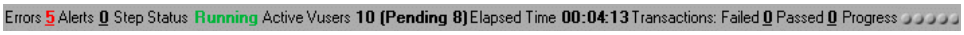
During session execution, you can use the Vuser Management window in the Execute view to monitor the actions of all the Vusers and Vuser groups in the session.

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 Mercury Tuning Console.
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.

Status	Description
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.



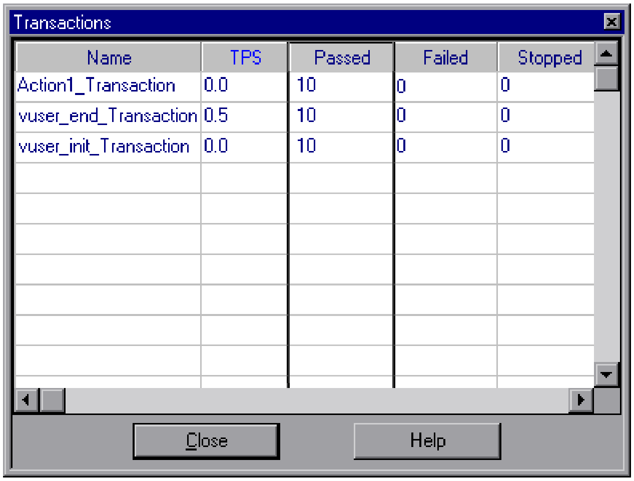
Errors 5 Alerts 0 Step Status Running Active Vusers 10 (Pending 8) Elapsed Time 00:04:13 Transactions: Failed 0 Passed 0 Progress

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 tool bar is a link. Click the link to open the Transactions window.



Name	TPS	Passed	Failed	Stopped
Action1_Transaction	0.0	10	0	0
vuser_end_Transaction	0.5	10	0	0
vuser_init_Transaction	0.0	10	0	0

Understanding the Transactions Window

The Transactions window lets you view details of individual transactions in a script.

Name: Transaction name.

TPS: Number of transactions per second.

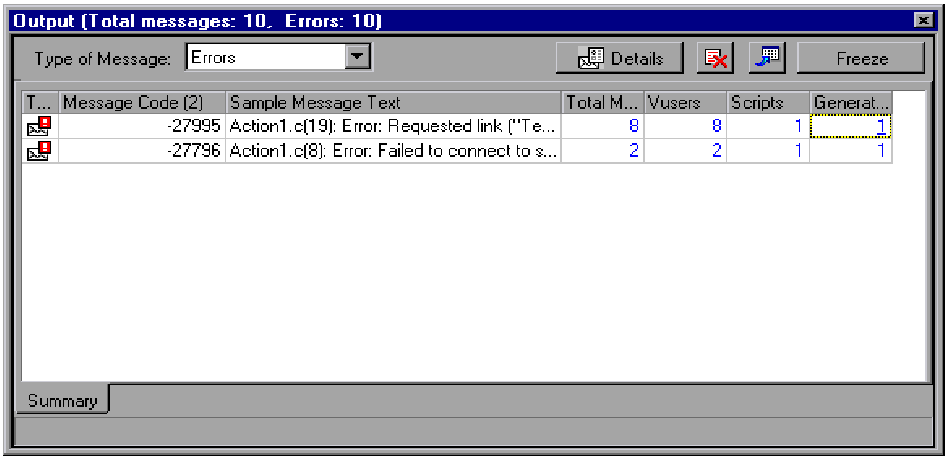
Passed: Number of transactions that passed.

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 tool bar 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.



Output (Total messages: 10, Errors: 10)

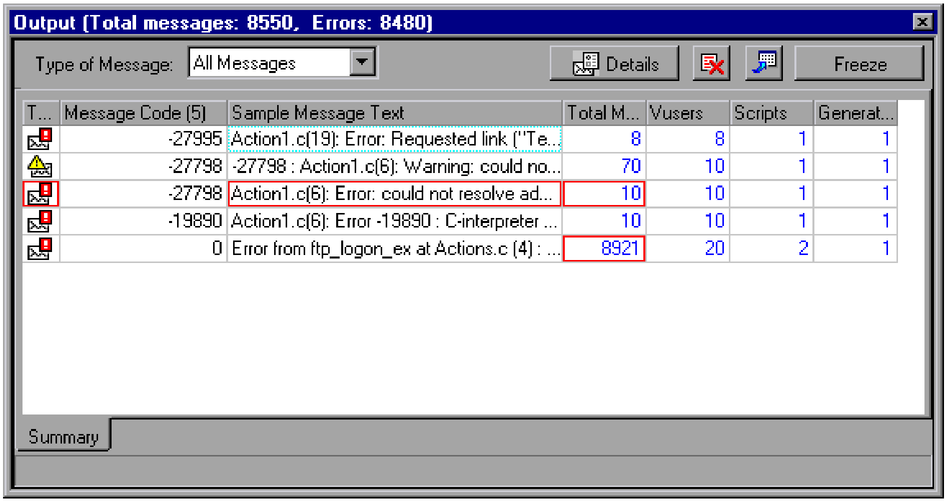
Type of Message: Errors

T...	Message Code (2)	Sample Message Text	Total M...	Vusers	Scripts	Generat...
	-27995	Action1.c(19): Error: Requested link ("Te...	8	8	1	1
	-27796	Action1.c(8): Error: Failed to connect to s...	2	2	1	1

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 230.

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.



Output (Total messages: 8550, Errors: 8480)

Type of Message: All Messages

T...	Message Code (5)	Sample Message Text	Total M...	Vusers	Scripts	Generat...
❗	-27995	Action1.c(19): Error: Requested link ("Te...	8	8	1	1
⚠	-27798	-27798 : Action1.c(6): Warning: could no...	70	10	1	1
❗	-27798	Action1.c(6): Error: could not resolve ad...	10	10	1	1
❗	-19890	Action1.c(6): Error -19890 : C-interpreter ...	10	10	1	1
❗	0	Error from ftp_logon_ex at Actions.c (4) : ...	8921	20	2	1

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.

Mercury Tuning Console 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 496 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 Mercury Tuning Console 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.

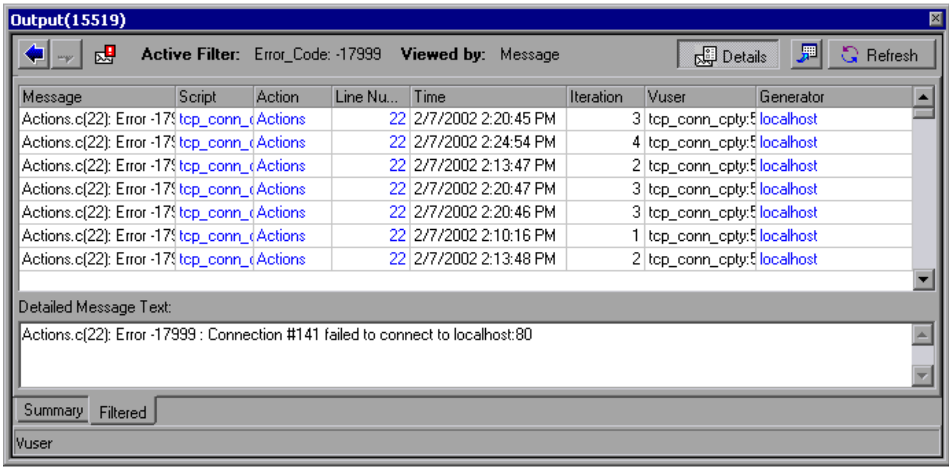
You can drill down further on the entries displayed in blue. Note that when you drill down on a Vuser, the Vuser log opens. When you drill down on a load generator, the Load Generators dialog box opens, displaying the load generator you selected. When you drill down on a script (or Action or Line Number), 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 498.

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.



Message	Script	Action	Line Nu...	Time	Iteration	Vuser	Generator
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:20:45 PM	3	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:24:54 PM	4	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:13:47 PM	2	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:20:47 PM	3	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:20:46 PM	3	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:10:16 PM	1	tcp_conn_cpty:5	localhost
Actions.c(22): Error -179	tcp_conn_c	Actions	22	2/7/2002 2:13:48 PM	2	tcp_conn_cpty:5	localhost

Detailed Message Text:

```
Actions.c(22): Error -17999 : Connection #141 failed to connect to localhost:80
```

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.

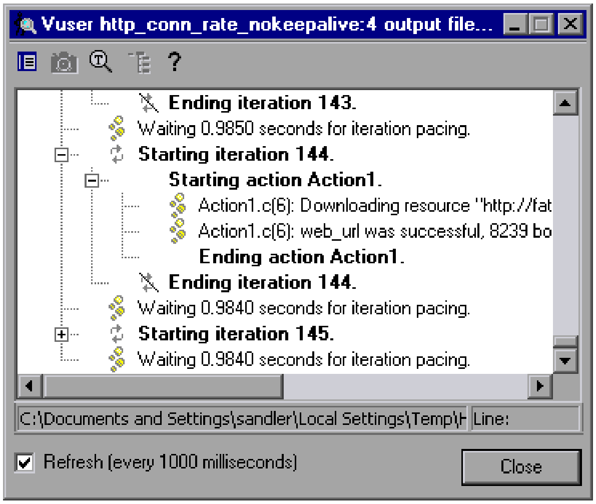
To move between the various drill-down levels, click the **Previous view** and **Next view** buttons in the upper left-hand corner of the Output window.

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.



```
Ending iteration 143.
Waiting 0.9850 seconds for iteration pacing.
Starting iteration 144.
  Starting action Action1.
    Action1.c(6): Downloading resource "http://fat
    Action1.c(6): web_url was successful, 8239 bo
    Ending action Action1.
  Ending iteration 144.
Waiting 0.9840 seconds for iteration pacing.
Starting iteration 145.
Waiting 0.9840 seconds for iteration pacing.
```

C:\Documents and Settings\sandler\Local Settings\Temp\F Line:

☑ Refresh (every 1000 milliseconds)

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 498.

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.

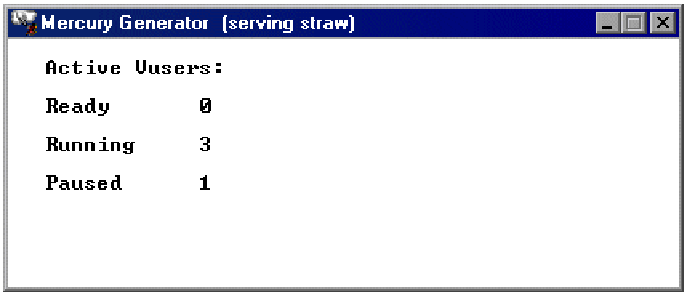
➤ To search the Vuser log for specific text, click the **Find Text** button, and enter the text you want to search for in the text box.

➤ 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 Agent icon on the task bar. Mercury Tuning Console displays a summary of the Ready, Running, and Paused Vusers.



Mercury Generator (serving straw)

```
Active Vusers:

Ready        0

Running      3

Paused       1
```

20

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

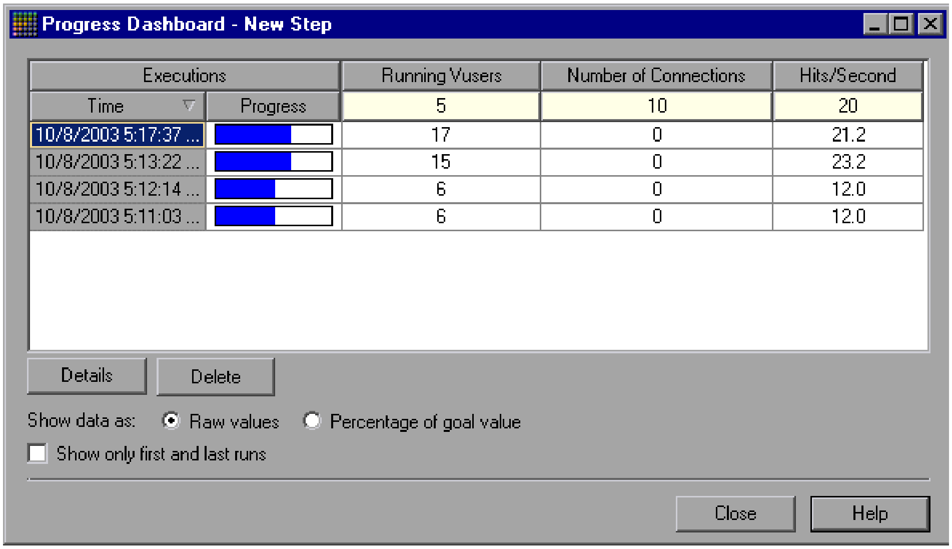
Viewing the Progress Dashboard

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.



Executions		Running Vusers	Number of Connections	Hits/Second
Time	Progress	5	10	20
10/8/2003 5:17:37 ...		17	0	21.2
10/8/2003 5:13:22 ...		15	0	23.2
10/8/2003 5:12:14 ...		6	0	12.0
10/8/2003 5:11:03 ...		6	0	12.0

Show data as: ● Raw values ○ Percentage of goal value

□ Show only first and last runs

Understanding the Progress Dashboard

The Progress Dashboard shows you how close each run of the benchmark step 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 22). Each row in the table shows the measurements that were achieved in a run of the benchmark step.

To view general information about the benchmark step run, select it and click **Details**. The Step Details dialog box is displayed. (See "Understanding the Step Details Dialog Box" on page 262.)

Delete: Removes the selected step run from the Progress Dashboard.

Executions: Lists all the benchmark 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 benchmark runs of the session.

21

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 Console 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 Mercury Tuning Console's knowledge base, and highlights the values that differ from the recommended ones

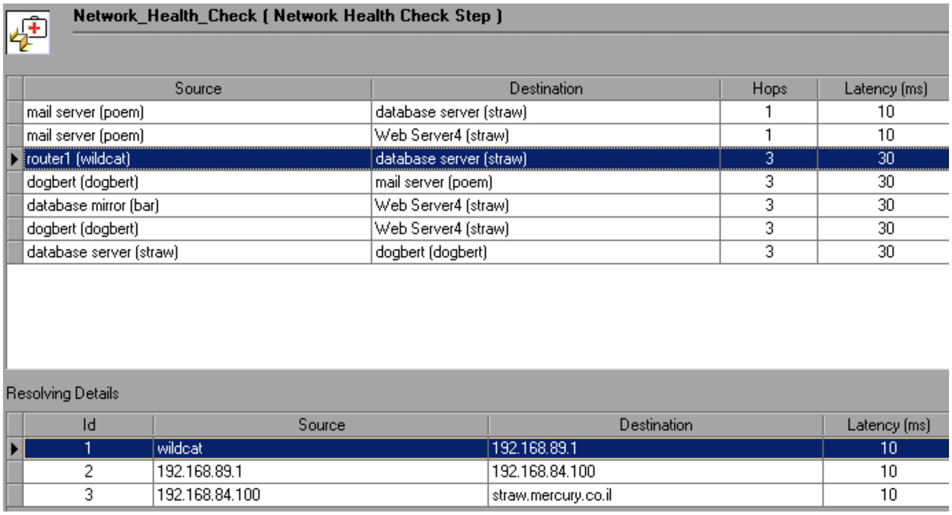
Network Health Check Step

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**. Mercury Tuning Console performs the step and displays the results, as in the following example.



Network_Health_Check (Network Health Check Step)

Source	Destination	Hops	Latency (ms)
mail server (poem)	database server (straw)	1	10
mail server (poem)	Web Server4 (straw)	1	10
router1 (wildcat)	database server (straw)	3	30
dogbert (dogbert)	mail server (poem)	3	30
database mirror (bar)	Web Server4 (straw)	3	30
dogbert (dogbert)	Web Server4 (straw)	3	30
database server (straw)	dogbert (dogbert)	3	30

Resolving Details

Id	Source	Destination	Latency (ms)
1	wildcat	192.168.89.1	10
2	192.168.89.1	192.168.84.100	10
3	192.168.84.100	straw.mercury.co.il	10

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, Mercury Tuning Console 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.

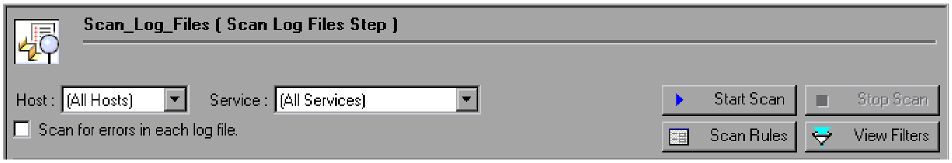
The upper section of the Scan Logs view consists of a table that displays information about log files scanned by Mercury Tuning Console.

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 252.

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.

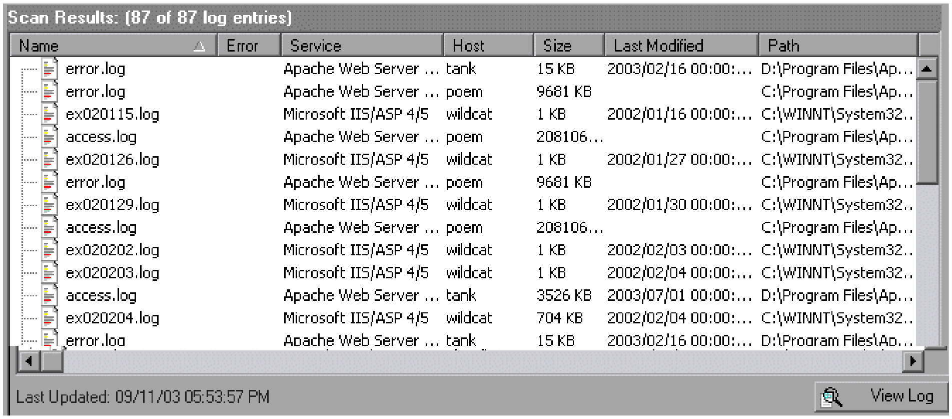


Scan_Log_Files (Scan Log Files Step)

Host : (All Hosts) Service : (All Services)

□ Scan for errors in each log file.

3. Click **Start Scan**. The Console displays a list of the hosts' log files.



Scan Results: (87 of 87 log entries)

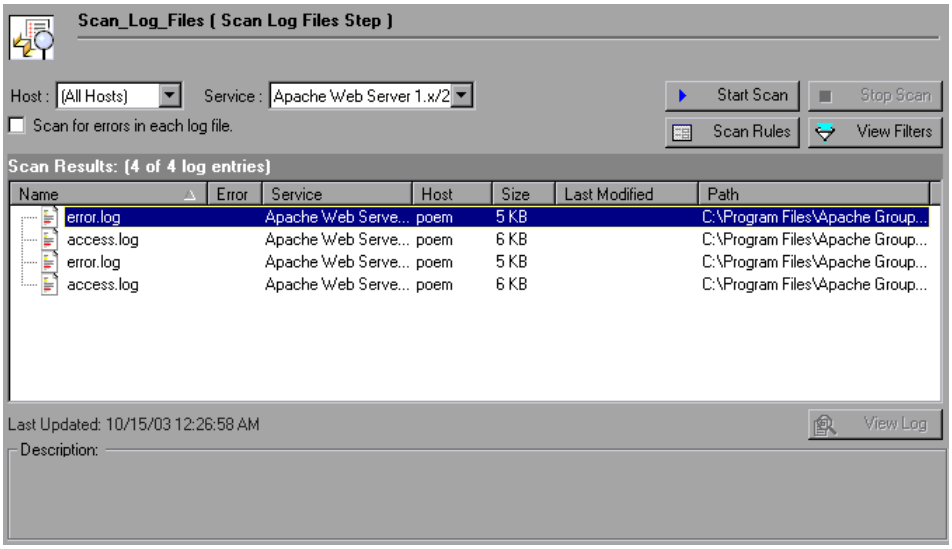
Name	Error	Service	Host	Size	Last Modified	Path
error.log		Apache Web Server ...	tank	15 KB	2003/02/16 00:00:...	D:\Program Files\Ap...
error.log		Apache Web Server ...	poem	9681 KB		C:\Program Files\Ap...
ex020115.log		Microsoft IIS/ASP 4/5	wildcat	1 KB	2002/01/16 00:00:...	C:\WINNT\System32..
access.log		Apache Web Server ...	poem	208106...		C:\Program Files\Ap...
ex020126.log		Microsoft IIS/ASP 4/5	wildcat	1 KB	2002/01/27 00:00:...	C:\WINNT\System32..
error.log		Apache Web Server ...	poem	9681 KB		C:\Program Files\Ap...
ex020129.log		Microsoft IIS/ASP 4/5	wildcat	1 KB	2002/01/30 00:00:...	C:\WINNT\System32..
access.log		Apache Web Server ...	poem	208106...		C:\Program Files\Ap...
ex020202.log		Microsoft IIS/ASP 4/5	wildcat	1 KB	2002/02/03 00:00:...	C:\WINNT\System32..
ex020203.log		Microsoft IIS/ASP 4/5	wildcat	1 KB	2002/02/04 00:00:...	C:\WINNT\System32..
access.log		Apache Web Server ...	tank	3526 KB	2003/07/01 00:00:...	D:\Program Files\Ap...
ex020204.log		Microsoft IIS/ASP 4/5	wildcat	704 KB	2002/02/04 00:00:...	C:\WINNT\System32..
error.log		Apache Web Server ...	tank	15 KB	2003/02/16 00:00:...	D:\Program Files\Ap...

Last Updated: 09/11/03 05:53:57 PM

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.



Scan_Log_Files (Scan Log Files Step)

Host : (All Hosts) Service : Apache Web Server 1.x/2

□ Scan for errors in each log file.

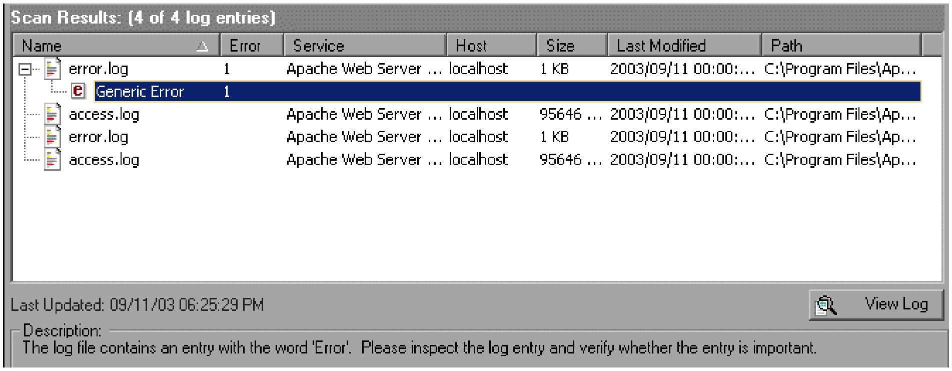
Scan Results: (4 of 4 log entries)

Name	Error	Service	Host	Size	Last Modified	Path
error.log		Apache Web Serve...	poem	5 KB		C:\Program Files\Apache Group...
access.log		Apache Web Serve...	poem	6 KB		C:\Program Files\Apache Group...
error.log		Apache Web Serve...	poem	5 KB		C:\Program Files\Apache Group...
access.log		Apache Web Serve...	poem	6 KB		C:\Program Files\Apache Group...

Last Updated: 10/15/03 12:26:58 AM

Description:

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.



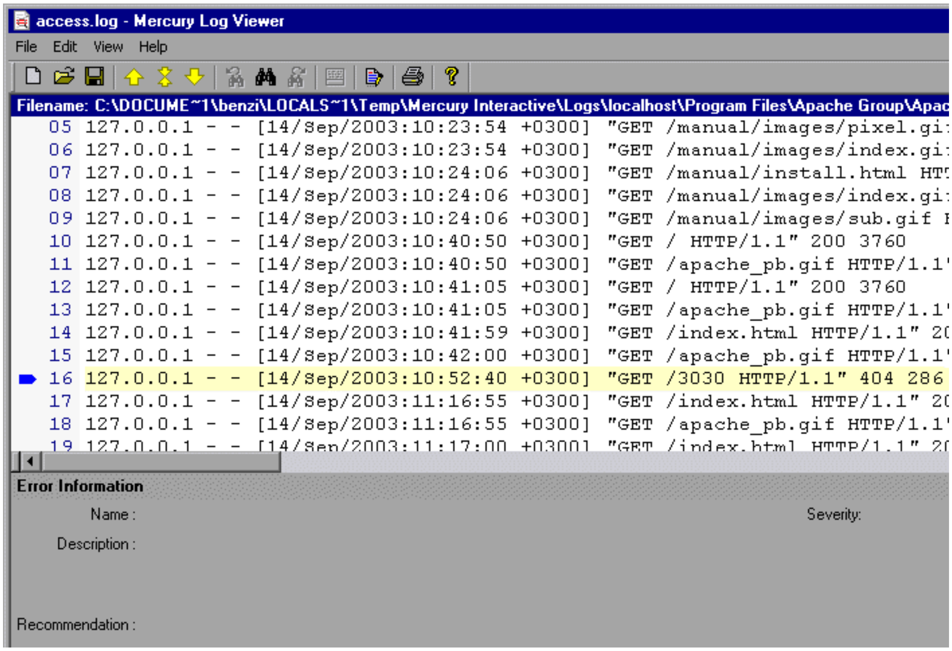
Scan Results: (4 of 4 log entries)

Name	Error	Service	Host	Size	Last Modified	Path
error.log	1	Apache Web Server ...	localhost	1 KB	2003/09/11 00:00:...	C:\Program Files\Ap...
Generic Error	1					
access.log		Apache Web Server ...	localhost	95646 ...	2003/09/11 00:00:...	C:\Program Files\Ap...
error.log		Apache Web Server ...	localhost	1 KB	2003/09/11 00:00:...	C:\Program Files\Ap...
access.log		Apache Web Server ...	localhost	95646 ...	2003/09/11 00:00:...	C:\Program Files\Ap...

Last Updated: 09/11/03 06:25:29 PM

Description:
The log file contains an entry with the word 'Error'. Please inspect the log entry and verify whether the entry is important.

5 To view the error message in the log file, click **View Log**. The log viewer opens. The error message entries are highlighted.



access.log - Mercury Log Viewer

Filename: C:\DOCUME~1\benzi\LOCALS~1\Temp\Mercury Interactive\Logs\localhost\Program Files\Apache Group\Apac

```
05 127.0.0.1 - - [14/Sep/2003:10:23:54 +0300] "GET /manual/images/pixel.gi:
06 127.0.0.1 - - [14/Sep/2003:10:23:54 +0300] "GET /manual/images/index.gi:
07 127.0.0.1 - - [14/Sep/2003:10:24:06 +0300] "GET /manual/install.html HT
08 127.0.0.1 - - [14/Sep/2003:10:24:06 +0300] "GET /manual/images/index.gi:
09 127.0.0.1 - - [14/Sep/2003:10:24:06 +0300] "GET /manual/images/sub.gif H
10 127.0.0.1 - - [14/Sep/2003:10:40:50 +0300] "GET / HTTP/1.1" 200 3760
11 127.0.0.1 - - [14/Sep/2003:10:40:50 +0300] "GET /apache_pb.gif HTTP/1.1'
12 127.0.0.1 - - [14/Sep/2003:10:41:05 +0300] "GET / HTTP/1.1" 200 3760
13 127.0.0.1 - - [14/Sep/2003:10:41:05 +0300] "GET /apache_pb.gif HTTP/1.1'
14 127.0.0.1 - - [14/Sep/2003:10:41:59 +0300] "GET /index.html HTTP/1.1" 20
15 127.0.0.1 - - [14/Sep/2003:10:42:00 +0300] "GET /apache_pb.gif HTTP/1.1'
16 127.0.0.1 - - [14/Sep/2003:10:52:40 +0300] "GET /3030 HTTP/1.1" 404 286
17 127.0.0.1 - - [14/Sep/2003:11:16:55 +0300] "GET /index.html HTTP/1.1" 20
18 127.0.0.1 - - [14/Sep/2003:11:16:55 +0300] "GET /apache_pb.gif HTTP/1.1'
19 127.0.0.1 - - [14/Sep/2003:11:17:00 +0300] "GET /index.html HTTP/1.1" 20
```

Error Information

Name :

Severity:

Description :

Recommendation :

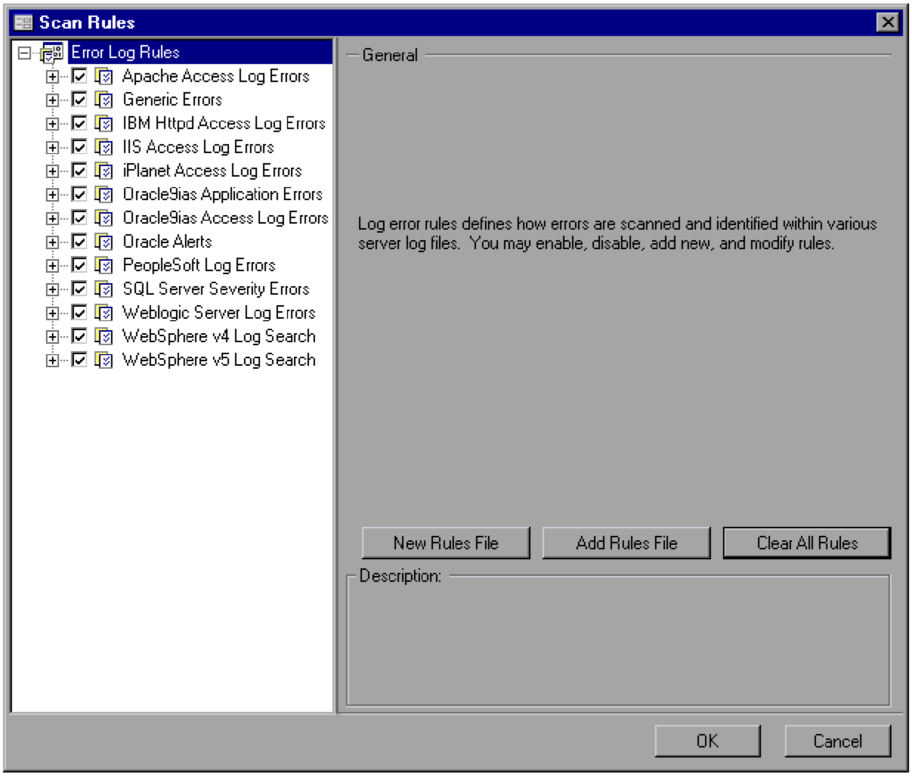
6 Use the up and down arrows to move to the next or previous errors.

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.



Error Log Rules
- ☑ Apache Access Log Errors
- ☑ Generic Errors
- ☑ IBM Httpd Access Log Errors
- ☑ IIS Access Log Errors
- ☑ iPlanet Access Log Errors
- ☑ Oracle9ias Application Errors
- ☑ Oracle9ias Access Log Errors
- ☑ Oracle Alerts
- ☑ PeopleSoft Log Errors
- ☑ SQL Server Severity Errors
- ☑ Weblogic Server Log Errors
- ☑ WebSphere v4 Log Search
- ☑ WebSphere v5 Log Search

Log error rules defines how errors are scanned and identified within various server log files. You may enable, disable, add new, and modify rules.

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.

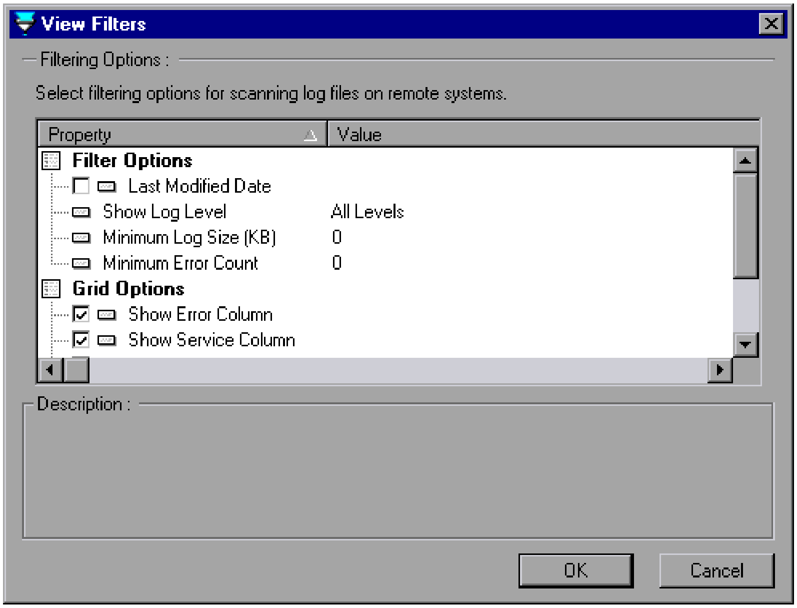
To import a rules file, click **Add Rules File** and choose the file you want to import.

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.



View Filters

Filtering Options :

Select filtering options for scanning log files on remote systems.

Property	Value
Filter Options	
□ Last Modified Date	
Show Log Level	All Levels
Minimum Log Size (KB)	0
Minimum Error Count	0
Grid Options	
☑ Show Error Column	
☑ Show Service Column	

Description :

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, Mercury Tuning Console scans the Apache log file on the Web server. When the scanning is complete, Mercury Tuning Console 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.

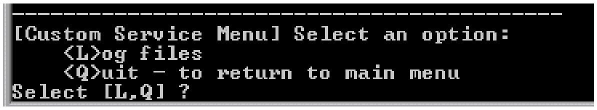
Description: A description of the selected log file, or the text of the selected error message.

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.

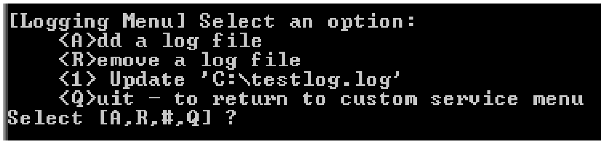
To define a custom service and specify the log file location:

1. Invoke the pe_registry utility (see "Using the Performance Tuner Registry" on page 438).
2. Type **L** and press <Enter> to list the current tuners.
3. Choose **Custom Service** and press <Enter>. The Custom Service menu opens.



```
-------------------------------------------
[Custom Service Menu] Select an option:
    <L>og files
    <Q>uit - to return to main menu
Select [L,Q] ?
```

4. Type **L** and press <Enter>. The Logging menu opens.



```
[Logging Menu] Select an option:
    <A>dd a log file
    <R>emove a log file
    <1> Update 'C:\testlog.log'
    <Q>uit - to return to custom service menu
Select [A,R,#,Q] ?
```

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 Mercury Tuning Console'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

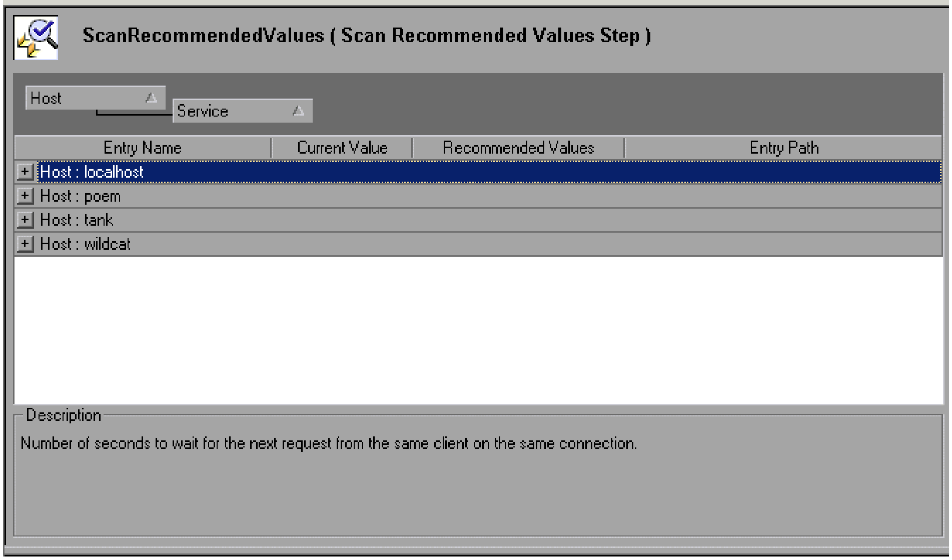
When you create a Scan Recommended Values step, Mercury Tuning Console displays a table showing the current and recommended values of the services running on the host machines, if there are any differences. If there are no differences between a host's current values and the recommended values for its 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.



ScanRecommendedValues (Scan Recommended Values Step)

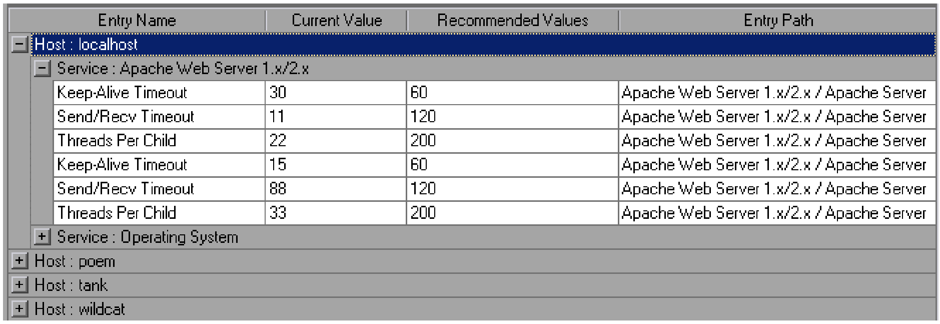
Host
Service

Entry Name	Current Value	Recommended Values	Entry Path
+ Host : localhost			
+ Host : poem			
+ Host : tank			
+ Host : wildcat			

Description

Number of seconds to wait for the next request from the same client on the same connection.

2 To view the services and their property values that differ, 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**.

   Mercury Tuning Console saves a report of the recommended values in a file with the name you specified.

22

Generating a Session Report

Mercury Tuning Console 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 Mercury Tuning Console 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

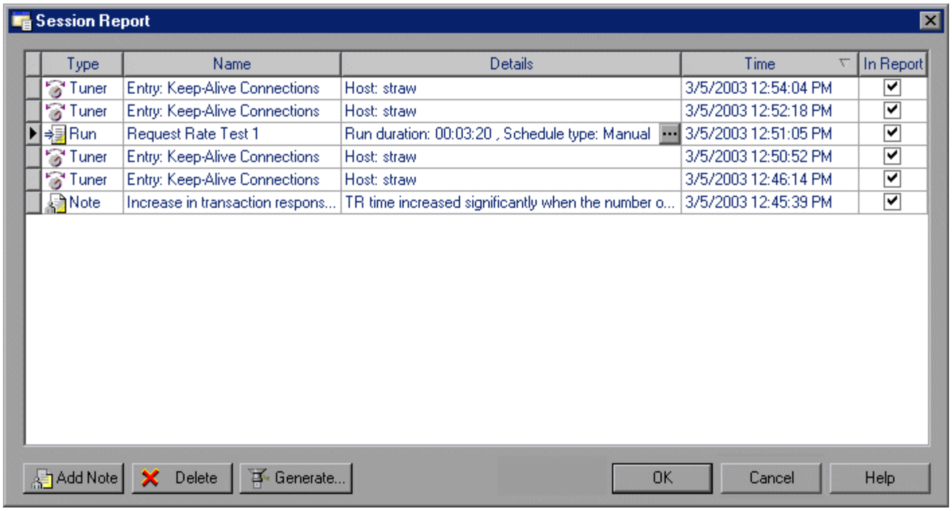
- Scheduler Information
- Scripts
- Run Time Settings
- Transaction Response Times
- Transaction Response Times Under Load
- Hardware Utilization - Windows Resources

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:

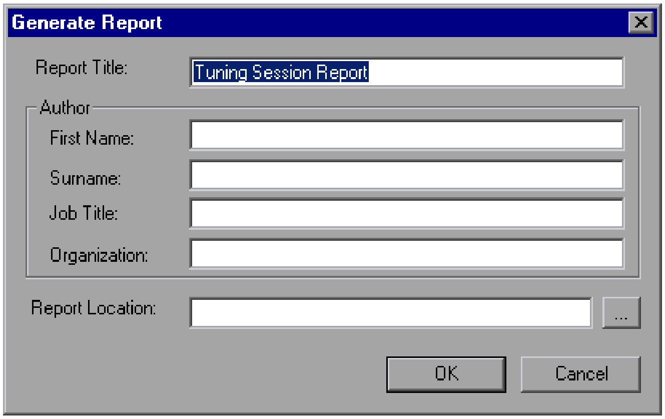
1 Click **Report**, or choose **Session** > **Session Report**.

The Session Report dialog box appears.



Type	Name	Details	Time	In Report
Tuner	Entry: Keep-Alive Connections	Host: straw	3/5/2003 12:54:04 PM	☑
Tuner	Entry: Keep-Alive Connections	Host: straw	3/5/2003 12:52:18 PM	☑
Run	Request Rate Test 1	Run duration: 00:03:20 , Schedule type: Manual	3/5/2003 12:51:05 PM	☑
Tuner	Entry: Keep-Alive Connections	Host: straw	3/5/2003 12:50:52 PM	☑
Tuner	Entry: Keep-Alive Connections	Host: straw	3/5/2003 12:46:14 PM	☑
Note	Increase in transaction respons...	TR time increased significantly when the number o...	3/5/2003 12:45:39 PM	☑

2 Click **Generate**. The Generate Report dialog box is displayed.



Generate Report

Report Title: Tuning Session Report

3 Enter information in the fields and click **OK**. Mercury Tuning Console 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 263.

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.

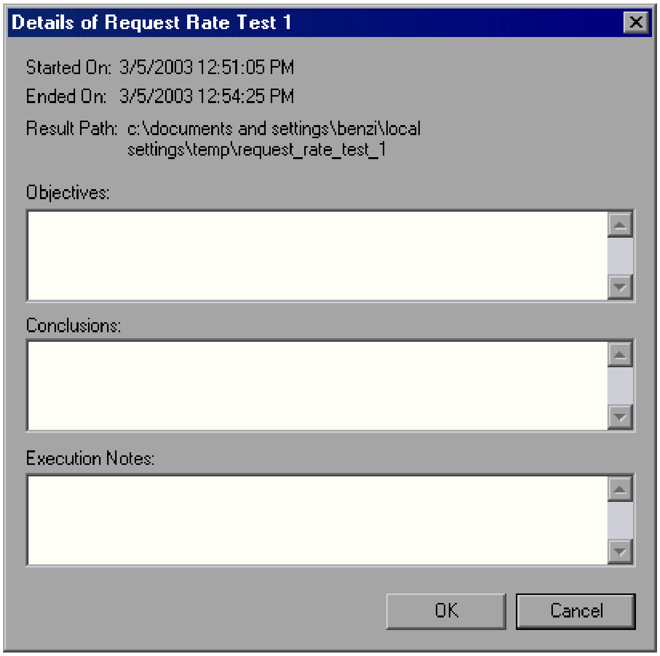
First Name, **Surname**, **Job Title** and **Organization:** Optional fields.

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.



Details of Request Rate Test 1

Started On: 3/5/2003 12:51:05 PM

Ended On: 3/5/2003 12:54:25 PM

Result Path: c:\documents and settings\benzi\local settings\temp\request_rate_test_1

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:

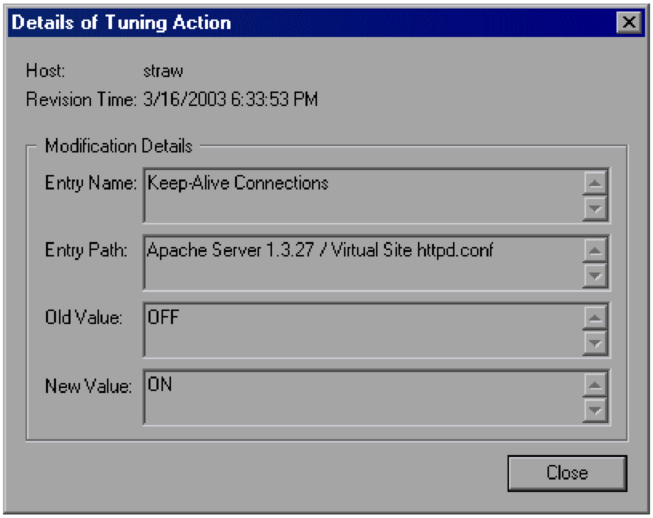
- **Started On**—When the step started executing
- **Ended On**—When the step execution ended
- **Result path**—Path to the directory where the step's results are stored
- **Objectives, Conclusions and Execution Notes**: You use these fields to enter relevant information about the step's execution.

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.



Details of Tuning Action

Host: straw

Revision Time: 3/16/2003 6:33:53 PM

Modification Details

Entry Name: Keep-Alive Connections

Entry Path: Apache Server 1.3.27 / Virtual Site httpd.conf

Old Value: OFF

New Value: ON

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

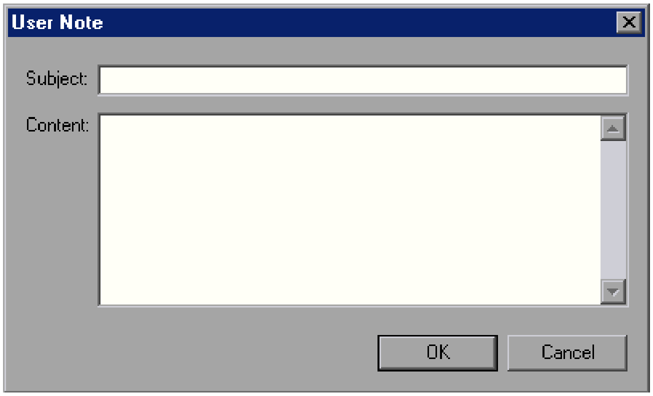
New Value—The property's value after the change

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. Mercury Tuning Console 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

23

Using Firewalls in Mercury Tuning Console

You can run Vusers and monitor your servers within a firewall, while the Console is outside of the firewall.

This chapter describes:

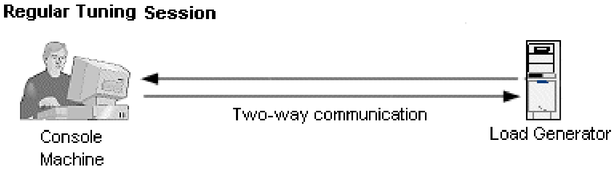
- Configuring Your System
- Overview of Running or Monitoring Vusers over a Firewall
- Checking Connectivity

About Using Firewalls in Mercury Tuning Console

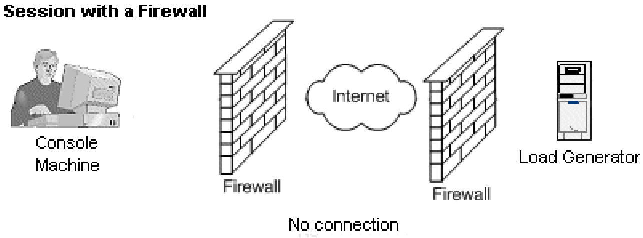
Working with a firewall means that you can prevent unauthorized access to or from a private network, on specific port numbers.

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 Mercury Tuning Console 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.

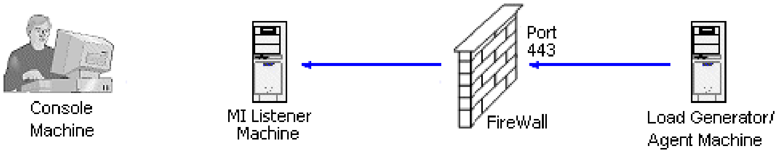


Mercury Tuning Console solves this problem by using a communication mechanism based on HTTPS or secured TCP/IP that uses the standard SSL port on the firewall (port 443). For more information on system configuration, see “Configuring Your System” on page 270.

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.

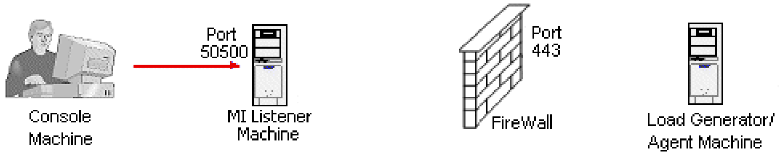
The MI Listener is a component that serves as a router between the Console and the LoadRunner agent.

Step 1: The LoadRunner Agent initiates a connection to the MI Listener through port 443



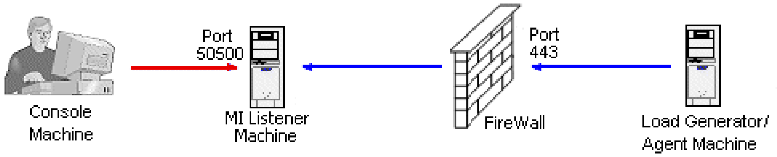
When the LoadRunner agent makes a connection to the MI Listener, the MI Listener keeps a listing of the connection to the agent using a symbolic name that the agent passed to it. When the Console connects to the MI Listener, it communicates to the MI Listener through port 50500.

Step 2: The Console connects 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.

Step 3: The Console and the Agent are connected

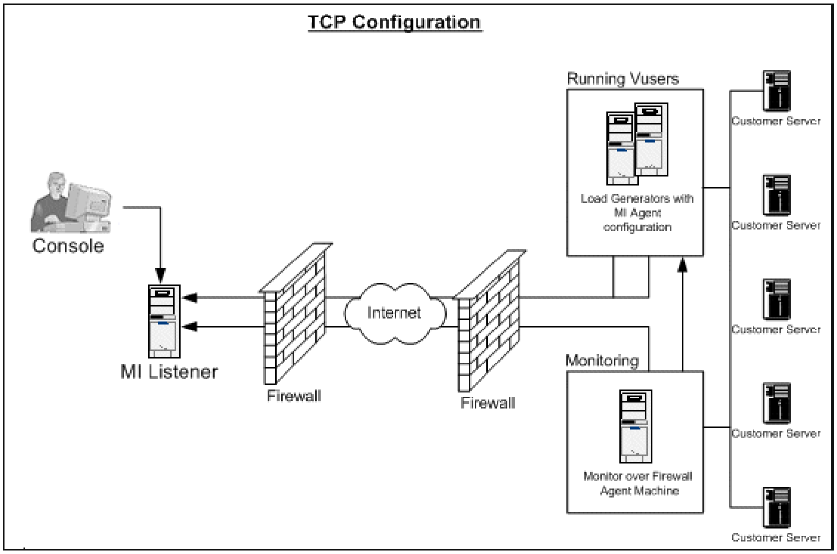


Configuring Your System

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.

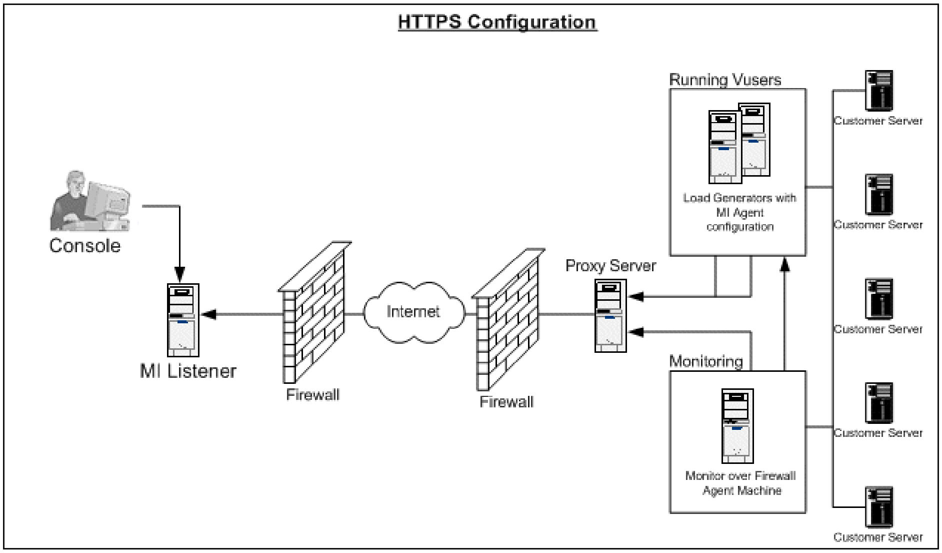
TCP Configuration

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.



Note: You can connect the Console to load generators over a firewall without using the MI Listener and LoadRunner agent. To do so, open port 54345 on a firewall in the load generator's LAN and in the Console's LAN to allow incoming and outgoing data.

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 Mercury Tuning Console 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 24, "Running Vusers Over a Firewall" and Chapter 25, "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 Mercury Tuning Console. 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 279.

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 280 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 289.

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 270 for information about where to install the Monitoring over the Firewall component. For configuration instructions, see "Installing Monitors over Firewall" on page 296.

5 Install the MI Listener on a machine outside the firewall.

For information about installing the MI Listener, refer to the *LoadRunner Installation Guide*. See the diagrams in "Configuring Your System" on page 270 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 290.

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 292.

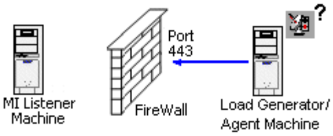
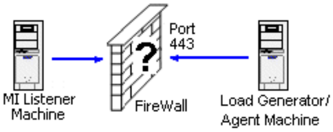
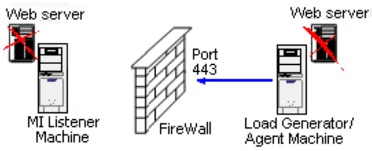
8 Configure Server Monitor Properties (monitoring over a firewall only).

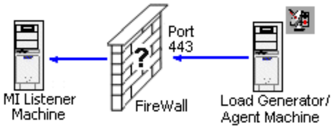
Configure the Server Monitor Properties and measurement frequency. See "Configuring Server Monitor Properties" on page 297, "Adding and Removing Measurements" on page 301, and "Configuring Measurement Frequency" on page 301.

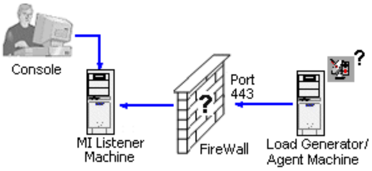
Checking Connectivity

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:  MI Listener Machine Port 443 FireWall Load Generator/ 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 280.
To check that port 443 is open:  MI Listener Machine Port 443 ? FireWall Load Generator/ Agent Machine	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:  Web server MI Listener Machine Port 443 FireWall Web server Load Generator/ Agent Machine	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:  MI Listener Machine ? Port 443 FireWall Load Generator/ Agent Machine	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 280. • Run the agent as a Process. Launch **<LoadRunner Installation>\Launch_service\bin\magentproc.exe**. If this works, this indicates an authentication issue with the LoadRunner Agent Service. Browse to the Service > LoadRunner 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  Console MI Listener Machine Port 443 ? FireWall ? Load Generator/ Agent Machine	• Check that you entered the servers that you want to monitor in the Monitor Configuration dialog box. (See “Configuring Server Monitor Properties” on page 297). • Start the LoadRunner Agent Process on the mediator machine. (See “Configuring LoadRunner Agents Inside the Firewall” on page 280). • 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 292). • 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 installation**>**\launch_service\bin\magnetproc.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.

For additional firewall troubleshooting, see “Troubleshooting Firewalls” on page 481.

24

Running Vusers Over a Firewall

The Mercury Tuning Console enables you to run Vusers over a firewall while the Console is outside of the firewall.

This chapter describes:

- 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 Mercury Tuning Console for running Vusers over the firewall, perform all the procedures in “Overview of Running Vusers Over a Firewall” on page 278. To enable monitoring of your servers from outside the firewall, you must also complete the procedures in “Monitoring Over a Firewall” on page 295.

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 272.

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 Mercury Tuning Console. 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 279.

2. **Configure the LoadRunner agent to operate over the firewall.**

   See “Configuring LoadRunner Agents Inside the Firewall” on page 280 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 289.

4. **Install the MI Listener on a machine outside the firewall.**

   For installation information, refer to the *LoadRunner Installation Guide*.

5. **Configure the security attributes on each MI Listener machine.**

   See “Installing and Configuring the MI Listener Outside the Firewall” on page 290.

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 292.

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 **Load Generator** installation from setup. Click **Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner Agent Service/Process** to check whether it is installed. If **Agent Service** or **Agent Process** appears on the list of LoadRunner options, then the agent was already installed.

If there is no agent installed, install the Load Generator component from the LoadRunner installation CD on the machine(s) running Vusers inside the firewall. See the diagrams in "Configuring Your System" on page 270 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 296. See the diagrams in "Configuring Your System" on page 270 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.

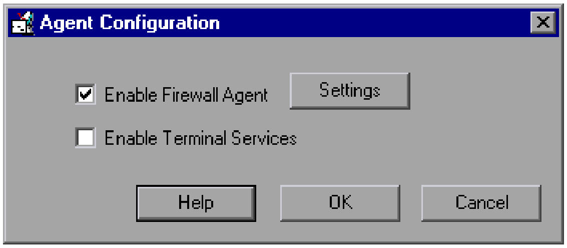
Configuring and Running the Windows LoadRunner Agent

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** > **Mercury LoadRunner** > **Advanced Settings**, or run **<LoadRunner root>\launch_service\bin\AgentConfig.exe**.

3 Select the **Enable Firewall Agent** check box, and then click **Settings**.

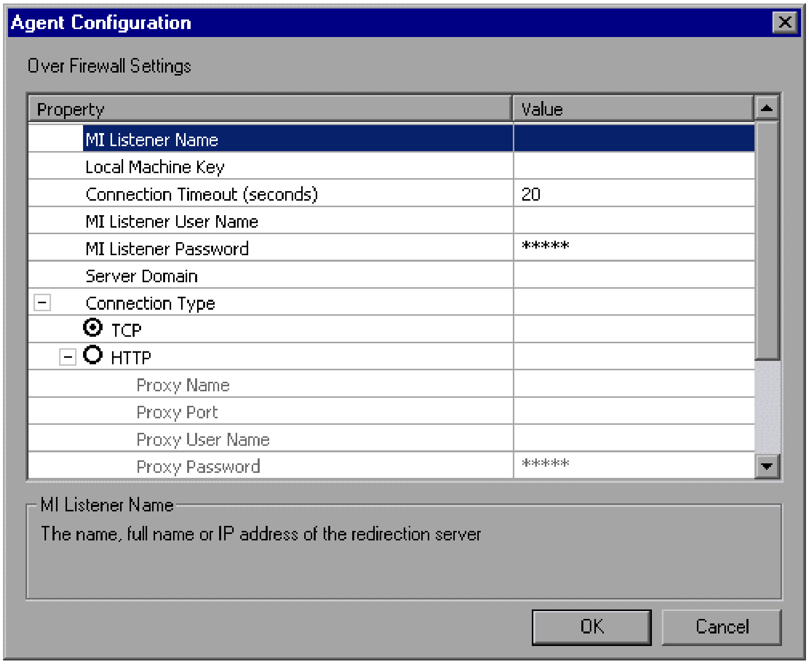


Agent Configuration

☑ Enable Firewall Agent

☐ Enable Terminal Services

The Agent Configuration dialog box opens.



Agent Configuration

Over Firewall Settings

Property	Value
MI Listener Name	
Local Machine Key	
Connection Timeout (seconds)	20
MI Listener User Name	
MI Listener Password	*****
Server Domain	
− Connection Type	
● TCP	
− ○ HTTP	
Proxy Name	
Proxy Port	
Proxy User Name	
Proxy Password	*****

MI Listener Name

The name, full name or IP address of the redirection server

4 Set each option as described in “Agent Configuration Settings” on page 286.

5 Click **OK** to save your changes or **Cancel** to cancel them.

6 Restart the LoadRunner agent by double-clicking the shortcut on the desktop, or from **Start** > **Programs** > **Mercury LoadRunner** > **LoadRunner 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 (see “Checking Connectivity” on page 274 for troubleshooting instructions).

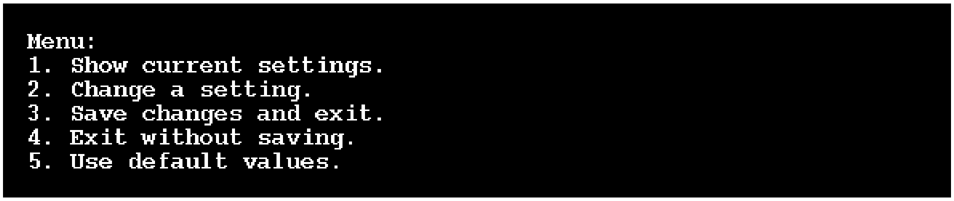
Configuring and Running the UNIX LoadRunner Agent

To configure the LoadRunner agent on UNIX machines:

1 Open **<LoadRunner root folder>/dat/br_lnch_server.cfg** in a text editor.

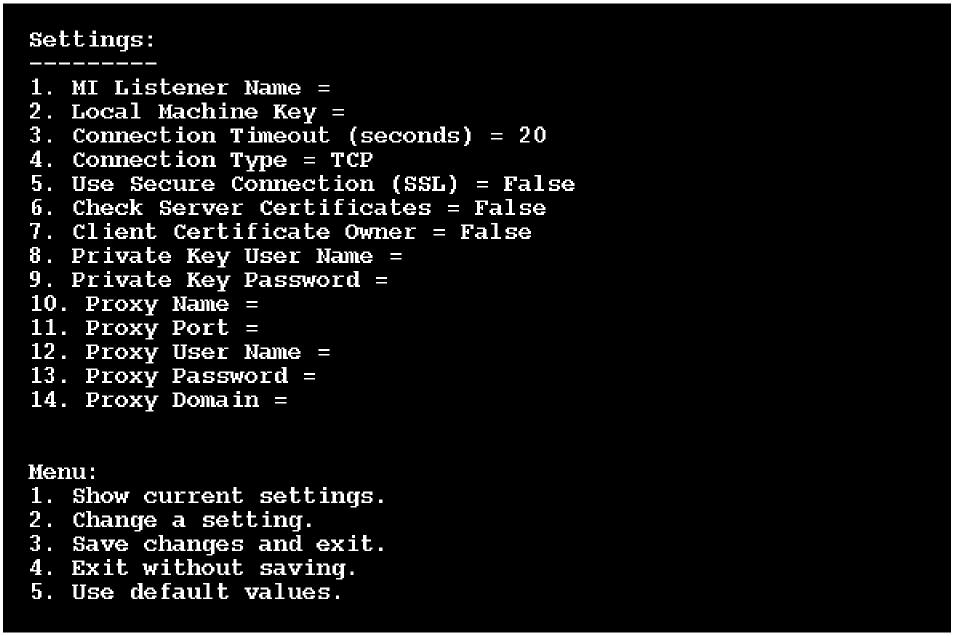
2 In the Firewall section, set FireWallServiceActive to 1 and save your changes.

3 Run **agent_config** from the **<LoadRunner root folder>/bin** directory to display the following menu:



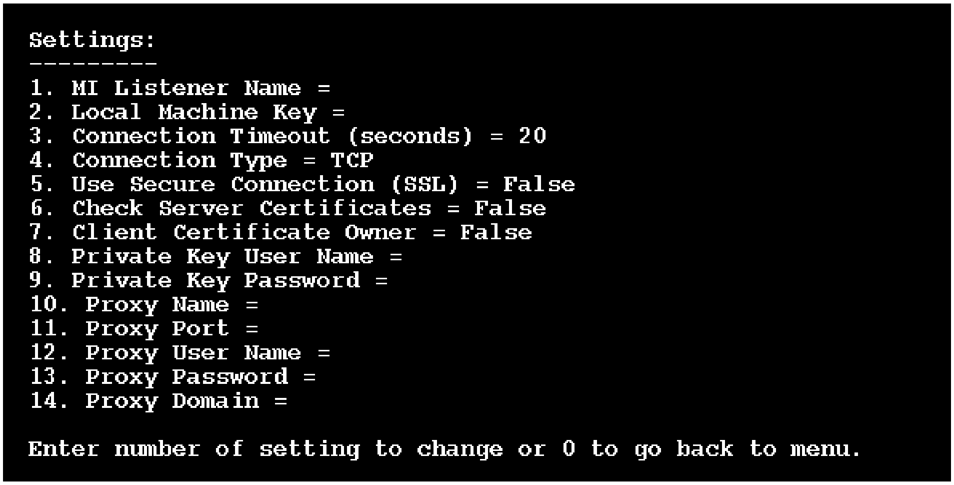
```
Menu:
1. Show current settings.
2. Change a setting.
3. Save changes and exit.
4. Exit without saving.
5. Use default values.
```

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:



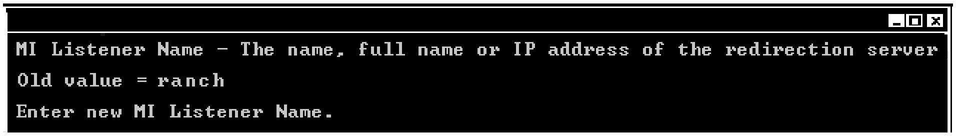
```
Settings:
---------
1. MI Listener Name =
2. Local Machine Key =
3. Connection Timeout (seconds) = 20
4. Connection Type = TCP
5. Use Secure Connection (SSL) = False
6. Check Server Certificates = False
7. Client Certificate Owner = False
8. Private Key User Name =
9. Private Key Password =
10. Proxy 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 286.

Examples of Changing Agent Settings in Unix

To change the MI Listener Name:

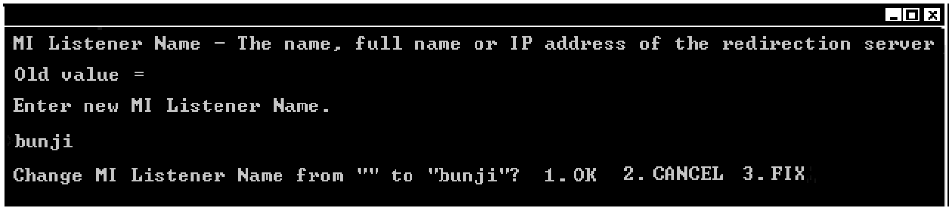
1 Enter 1 in the Settings menu to display the following screen:



```
MI Listener Name - The name, full name or IP address of the redirection server
Old value = ranch
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:



```
MI Listener Name - The name, full name or IP address of the redirection server
Old value =
Enter new MI Listener Name.
bunji
Change MI Listener Name from "" to "bunji"?  1.OK  2.CANCEL  3.FIX
```

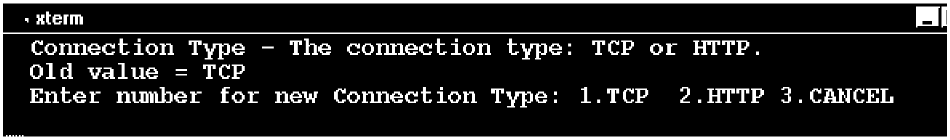
3 To keep the new value and return to the menu, enter 1.

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:

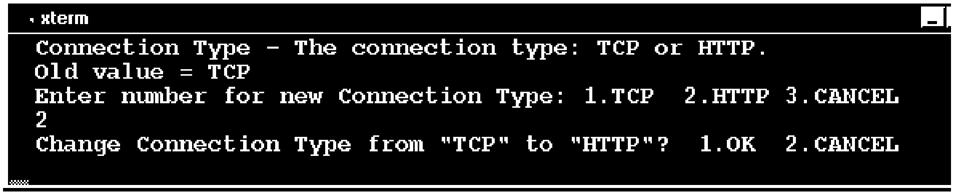
1 Enter 4 in the Settings menu to display the following screen:



```
Connection Type - The connection type: TCP or HTTP.
Old value = TCP
Enter number for new Connection Type: 1.TCP  2.HTTP 3.CANCEL
```

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:



```
Connection Type - The connection type: TCP or HTTP.
Old value = TCP
Enter number for new Connection Type: 1.TCP  2.HTTP 3.CANCEL
2
Change Connection Type from "TCP" to "HTTP"?  1.OK  2.CANCEL
```

3 To keep the new value and return to the menu, enter 1.

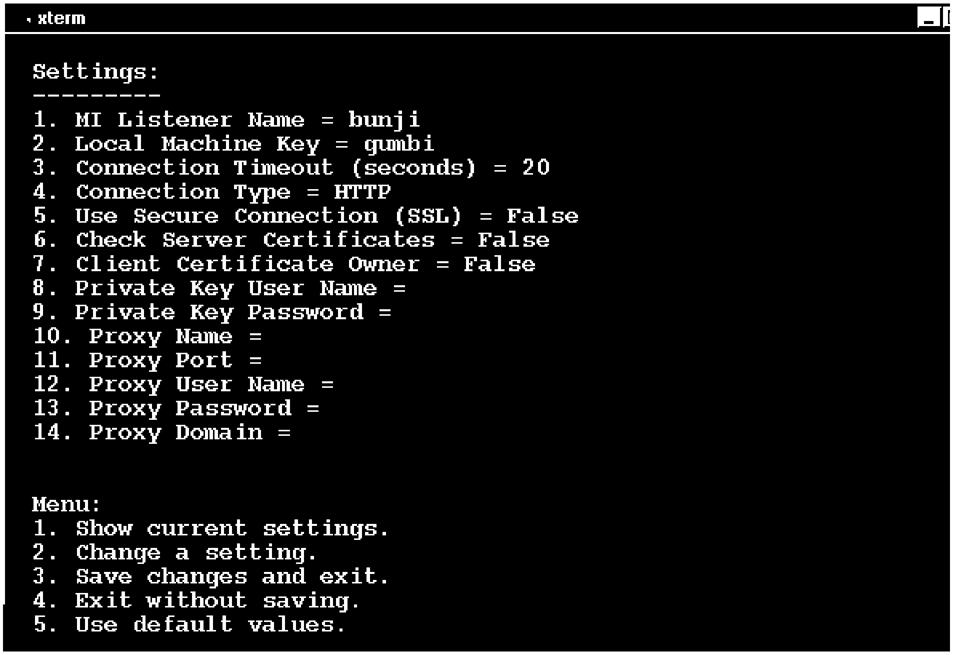
To discard the new value and return to the menu, enter 2.

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:



```
Settings:
---------
1. MI Listener Name = bunji
2. Local Machine Key = gumbi
3. Connection Timeout (seconds) = 20
4. Connection Type = HTTP
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.
```

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 (as described in “Agent Configuration Settings” on page 286), 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 root folder**>/**bin** directory.
2. To remove the LoadRunner agent, run the command m_daemon_setup -remove from the <**LoadRunner 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.

Option	Default Value	Description
MI Listener User Name	none	The username needed to connect to the MI Listener machine.
MI Listener 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.
Connection Type - TCP	TCP	Choose either **TCP** or **HTTP**, depending on the configuration you are using.
Connection Type - HTTP 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.**
Connection Type - HTTP Proxy Port	<IE proxy server name> or None	The proxy server connection port. This option is mandatory if the **Connection Type** option is set to **HTTP**.
Connection Type - HTTP Proxy User Name	None	The username of a user with connection rights to the proxy server.
Connection Type - HTTP Proxy Password	None	The user's password.
Connection Type - HTTP Proxy Domain	None	The user's domain if defined in the proxy server configuration. This option is required only if NTLM is used.
Use Secure Connection (SSL)	Disabled	Enable to connect using the Secure Sockets Layer protocol.

Option	Default Value	Description
Use Secure Connection (SSL) - 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**.
Use Secure Connection (SSL) - 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 enabled.
Use Secure Connection (SSL) - Use Client Certificate	Disabled	Enable to load the SSL certificate (if required by the server to allow the connection to be made). This option is relevant only if the **Use Secure Connection** option is enabled.

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.

Note: You can connect the Console to load generators over a firewall without using the MI Listener and LoadRunner agent. To do so, open port 54345 on a firewall in the Console's LAN and in the load generator's LAN to allow incoming and outgoing data.

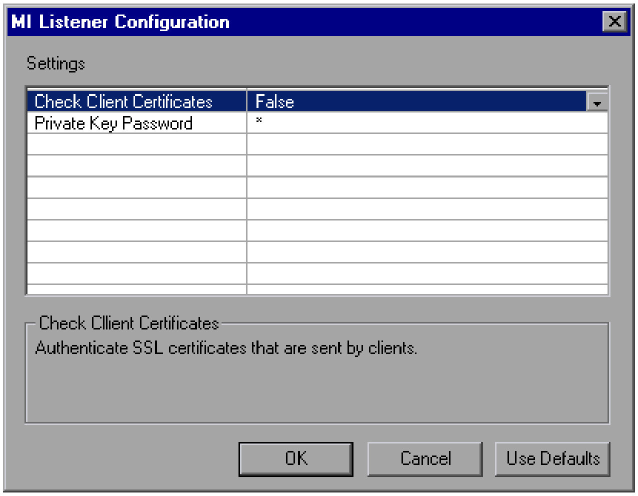
Installing and Configuring the MI Listener Outside the Firewall

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 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** > **Mercury LoadRunner** > **Advanced Settings**, or run **<LoadRunner root folder>\launch_service\bin\MILsnConfig.exe**.



MI Listener Configuration

Settings

Check Client Certificates	False
Private Key Password	*

Check Cllient Certificates

Authenticate SSL certificates that are sent by clients.

4 Set each option as described in "MI Listener Configuration Settings" on page 291.

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** > **Mercury LoadRunner**.

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.

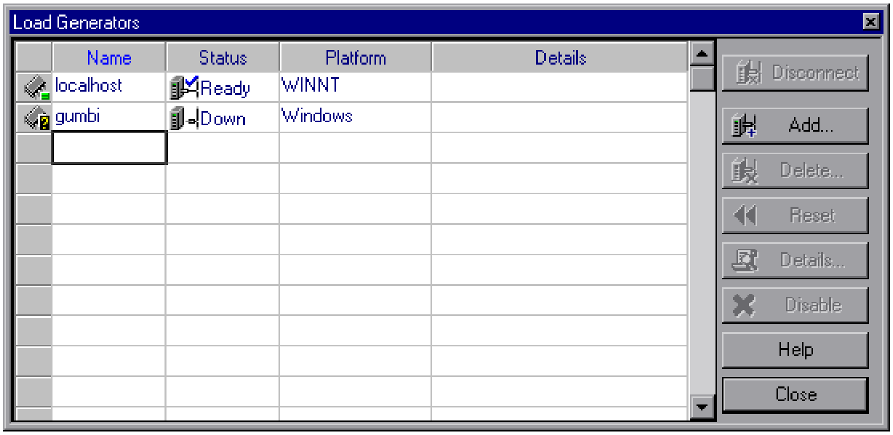
Configuring the Console to Run or Monitor Vusers Over a Firewall

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:

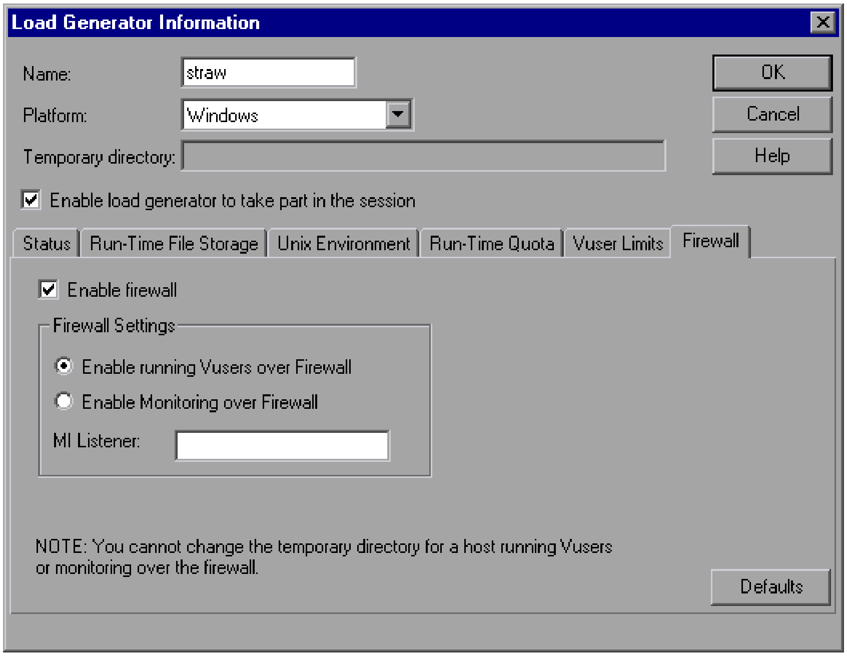
1. Create a new session or load an existing one.
2. Click the **Load Generators** icon to display the Load Generators window. In the Name field, enter the symbolic name of the server. This is the same name that you entered in the Local Machine Key setting in the Agent Configuration. (See “Agent Configuration Settings” on page 286). In the example below, the server name is gumbi.

   If the server is a UNIX server, change the Platform field to **UNIX**.



Name	Status	Platform	Details
localhost	Ready	WINNT	
gumbi	Down	Windows	

3 Select the Load Generator, and click **Details** to display the Load Generator Information.



Name: straw

Platform: Windows

Temporary directory:

☑ Enable load generator to take part in the session

☑ Enable firewall

Firewall Settings

● Enable running Vusers over Firewall

○ Enable Monitoring over Firewall

MI Listener:

NOTE: You cannot change the temporary directory for a host running Vusers or monitoring over the firewall.

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 MI Listener Name setting of the Agent Configuration dialog box. 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 274. For other firewall troubleshooting, see “Troubleshooting Firewalls” on page 481.

25

Monitoring Over a Firewall

You can monitor your servers within a firewall while the Console is outside of the firewall.

This chapter describes:

- 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 278, 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 Mercury Tuning Console 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 272.

For information on running Vusers over a firewall, see "Running Vusers Over a Firewall" on page 277.

Installing Monitors over Firewall

Monitors over Firewall may have been installed during the LoadRunner installation. To check whether it is installed, click **Start** > **Programs** > **Mercury LoadRunner** > **Advanced Settings.** If the **Monitor Configuration** option appears on the list of LoadRunner options, then Monitors over Firewall was already installed, and you can proceed to "Preparing for Data Collection" on page 297.

If there is no installation, install Monitors over Firewall on the mediator machine using one of the following:

- Perform a custom installation of Mercury Tuning Console from the LoadRunner CD, choosing only the **Monitors over Firewall** option.
- Obtain the **Monitors over Firewall** file from the Mercury 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 Mercury Tuning Console, refer to the *LoadRunner 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 277 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 292.

2 Remember that in the Firewall tab of the Load Generator Information dialog box, you should check **Enable Monitoring over Firewall** and enter the IP address of the MI Listener machine in the MI Listener field.

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.

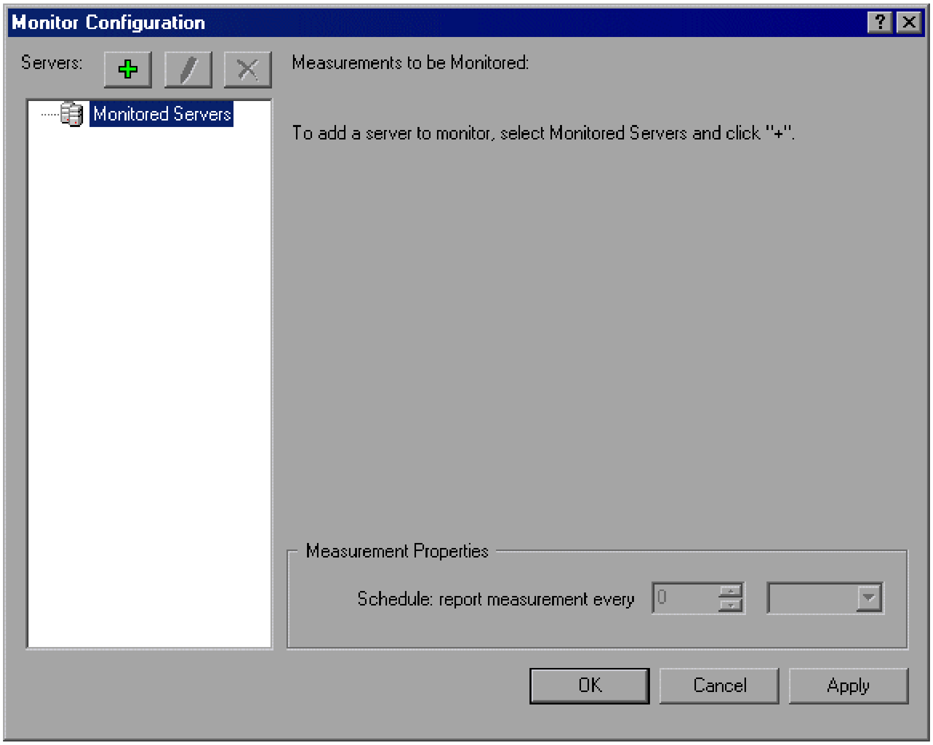
Configuring Server Monitor Properties

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:

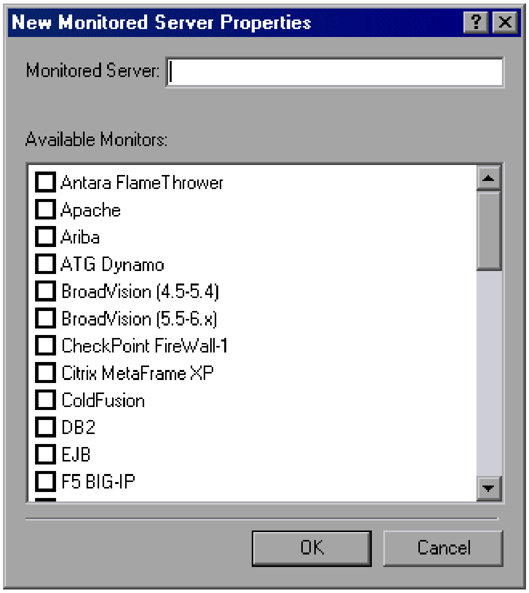
1 Select **Start** > **Programs** > **Mercury LoadRunner** > **Advanced Settings** > **Monitor Configuration**. For machines without the complete LoadRunner installation, select **Start** > **Programs** > **Server Monitor** > **Monitor Configuration.** The Monitor Configuration dialog box opens.



To add a server to monitor, select Monitored Servers and click "+".

Schedule: report measurement every 0

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.

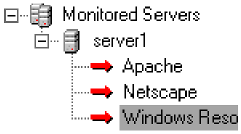
---

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 license key. To preview your license key information, click **Start** > **Programs** > **Mercury LoadRunner**. Mercury LoadRunner opens. Click the **License** button to display the LoadRunner License information.

---

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, Mercury Tuning Console displays default measurements in the Measurements to be Monitored section. You can specify the frequency at which Mercury Tuning Console reports the measurement in the Measurement Properties section. For details on selecting measurements, see “Adding and Removing Measurements” on page 301.

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 Mercury Tuning Console 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.

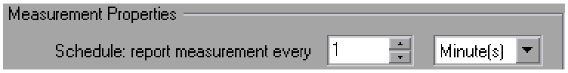
To remove a measurement from the measurements list:

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.



Measurement Properties

Schedule: report measurement every 1 Minute(s)

To set a measurement schedule for a measurement:

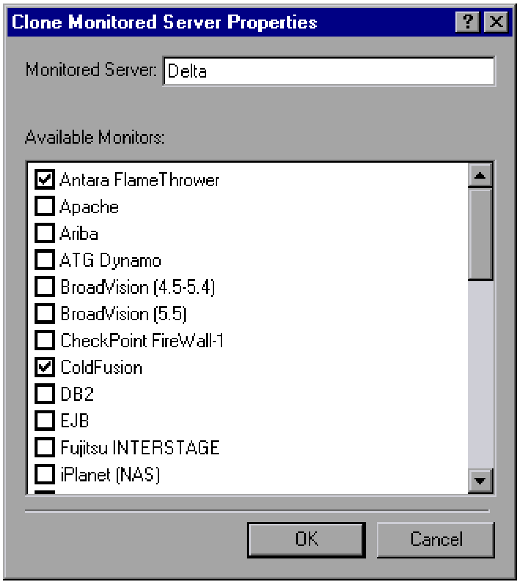
1. Select the configured server measurement you want to schedule.
2. Specify the frequency at which you want Mercury Tuning Console to report the measurement.
3. Click **Apply** to save your settings.

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.



Clone Monitored Server Properties

Monitored Server: Delta

Available Monitors:

☑ Antara FlameThrower
□ Apache
□ Ariba
□ ATG Dynamo
□ BroadVision (4.5-5.4)
□ BroadVision (5.5)
□ CheckPoint FireWall-1
☑ ColdFusion
□ DB2
□ EJB
□ Fujitsu INTERSTAGE
□ iPlanet (NAS)

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 (refer to “Configuring the Network Delay Time Monitor” in the *LoadRunner Monitor Reference*), 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 286.)

For example: 12.12.12.3:vds

Part V

Working with Diagnostics

26

J2EE Diagnostics Module

The J2EE Diagnostics module provides a monitor that traces, times, and troubleshoots individual transactions through J2EE Web, application, and database servers. It also enables you to quickly pinpoint problem servlets and JDBC calls to maximize business process performance, scalability, and efficiency.

The J2EE Diagnostics module monitor comes with a set of customized Analysis reports. Using these reports, you can communicate your insights to developers and managers, ultimately accelerating problem resolution, while avoiding expensive hardware upgrades.

This chapter describes:

- J2EE Diagnostics Module Architecture
- Setting up the Monitoring Environment
- Configuring J2EE Diagnostics on the Client Machine
- Viewing J2EE Diagnostics Data
- Examples of Modifying Application Server Configurations
- Troubleshooting the J2EE Diagnostics Module Monitor

About J2EE Diagnostics Monitoring

The J2EE Diagnostics module monitor provides the following information for each transaction:

- Average response time per method/query
- Number of method calls per second
- Breakdown information by layer, class, and method
- Chain of calls and call stack statistics

With this level of detail, you can get an overview of the entire chain of activity within the system. At the same time, you can break down J2EE layers into classes and methods to enable you to pinpoint the exact location where time is consumed. You can also view the transaction call chain to track the percent of time spent for each part of the transaction.

You can correlate the end user response time with the Web server activity (Servlets and JSPs data), application server activity (JNDI), and back-end activity of database requests (JDBC methods and SQL queries).

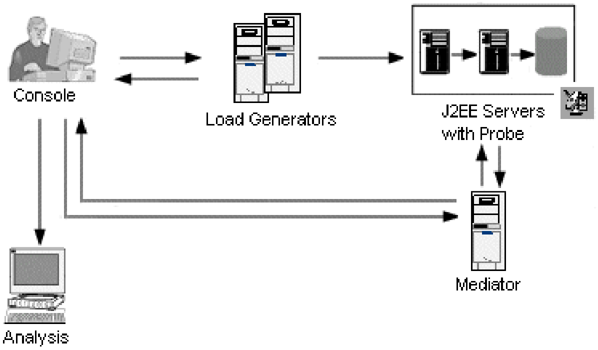
You can correlate the end user response time with the Web server activity (Servlets and JSPs data), application server activity (JNDI), and back-end activity of database requests (JDBC methods and SQL queries).

The J2EE Diagnostics module monitor enables you to analyze J2EE component metrics during a scenario or step run by using an agent—installed on the application server—to collect information on the J2EE components. These measurements are sent from the application server back to the Console through a channel contained in the J2EE Diagnostics module monitor. The J2EE Diagnostics module monitor supports the leading application servers, such as IBM WebSphere and BEA WebLogic. For information about the supported application servers, see the "Support Matrix" on page 311.

Note: The J2EE Diagnostics module monitor requires MSXML 3.0 or later (included in Internet Explorer 6.0). You can install MSXML 3.0 from the Microsoft MSDN Web site (http://msdn.microsoft.com/library/default.asp?url=/downloads/list/xmlgeneral.asp).

J2EE Diagnostics Module Architecture

The J2EE Diagnostics architecture is composed of the following components:



Probe: The Probe resides on the application server. It gathers information about the monitored transactions and passes data to the mediator.

Mediator: The Mediator component is used to connect between the probe (on the application server) and the Console machine. The Mediator processes the transaction data, and then passes it to the Console. Note that the Mediator should be installed on a load generator machine that resides in the same LAN as the Console and Probe machines. If you are monitoring over a firewall, the Console can be outside the firewall provided you open port 54345 on the firewall in the Probe/Mediator LAN and in the Console's LAN to allow incoming and outgoing data.

Console: Displays the online transaction data.

Analysis: Displays detailed diagnostic graphs and reports. For more information about the diagnostic graphs, see the *LoadRunner Analysis User's Guide*.

Note: You can install the Mediator and Probe components on one machine, or you can distribute them over multiple machines. We do not recommend installing the Mediator on the application server machine, since this can affect the load test results.

Setting up the Monitoring Environment

To use the J2EE Diagnostics module monitor, you must first install the J2EE Probe and Mediator from the LoadRunner installation CD. For installation information, see the *LoadRunner Installation Guide*.

After installing the Probe and Mediator components, you configure the J2EE Diagnostics monitor on the application server. You then enable the J2EE Diagnostics (from the Console) and select the default measurements you want to display, before running the session step.

This section describes the following topics:

- Configuring the J2EE Diagnostics Module Monitor on the Application Server
- Initial Configuration Settings
- Configuring JDBC Information Retrieval

Support Matrix

The following servers are supported:

Application Server	Version	Platform
WebLogic	6.x; 7.0; 8.1	Windows; Solaris
WebSphere	4.x; 5.x	Windows; AIX
Oracle 9i	1.0.2.2	Windows; Solaris; AIX
JBoss	2.4.x;3.04	Windows; Solaris; AIX
Sun ONE Application Server	6; 6.5; 7	Windows; Solaris; AIX
Sun ONE Web Server	6	Windows; Solaris; AIX

Configuring the J2EE Diagnostics Module Monitor on the Application Server

To monitor J2EE objects, you configure the J2EE Diagnostics module monitor on the application server machine. This section explains these steps in general. For examples of how to perform these steps for different application servers, see “Examples of Modifying Application Server Configurations” on page 330.

Note: Before making any changes, back up all startup scripts.

This chapter uses <**mercprobe**> to denote the J2EE probe home directory.

To update the application server configuration:

1. Update the Java command line that starts the application server. Add the following parameter at the beginning of the line:

```
-Xbootclasspath/p:<mercprobe>/classes/boot
```

For example,

```
-Xbootclasspath/p:/bea/mercprobe/classes/boot
```

Note: If your JVM uses a JIT option, such as -hotspot or -classic, make sure that the -Xbootclasspath option is entered following the JIT option.

---

2. Restart the application server instance for which you configured the probe.

Note: Your site administrator may configure the application server using an alternative, site-specific method. If this is the case, the generic procedure may be sufficient for the administrator to understand what changes need to be made.

Initial Configuration Settings

The following settings are automatically configured during the installation of the J2EE Diagnostics Module monitor on the application server:

Hooking mechanism: The J2EE Diagnostics Module monitor uses the Mercury J2EE Transaction Breakdown Monitor Initializer and Java hooking library.

Operation mode: The J2EE Diagnostics Module monitor uses the Auto Discovery operating mode. In this mode, the system automatically discovers the J2EE components (Servlet, JSP, EJB, JNDI, 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 Diagnostics Module monitor aggregates the measured data according to the JDBC operation, for example: SELECT,UPDATE,CREATE. To modify this configuration, see "Configuring JDBC Information Retrieval" on page 313.

Note: For information about alternative configuration settings, contact Mercury Customer Support.

Configuring JDBC Information Retrieval

1 Open <**J2EE Transaction Breakdown Home Directory**>\etc\dispatcher.properties.

2 In the property sql.parsing.mode, enter one of the following:

- "1" to measure the JDBC 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. If the sql.mode property does not exist, this is the default mode.
- "3" to aggregate the measured data according to specific SQL statement (including the operation, the tables it acted on, and other parameters of this statement).

Note: SQL Statements that exceed 3000 characters in length are not supported.

Configuring J2EE Diagnostics on the Client Machine

To monitor J2EE transactions, you must connect between the application server (with the probe) and the mediator. The application server and the Console connect via the mediator.

The settings that you configure are per session step. All scripts in the session step will run under the same diagnostics configuration.

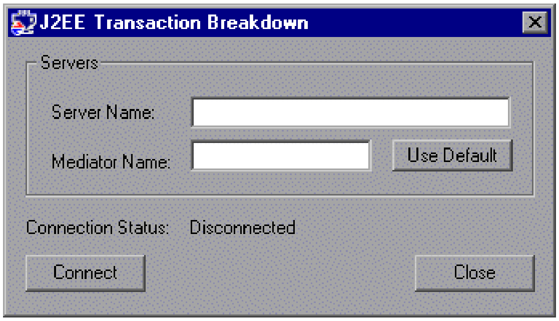
Note: Ensure that the application server has finished launching before attempting to connect the Console to the Probe/Mediator.

Configuring J2EE Transaction Breakdown on the Console

You can configure the transaction breakdown and reconnection settings using the J2EE Transaction Breakdown dialog box in the Console.

To define the connection on the Console:

1. From the Console menu, choose **J2EE** > **Connection**, or click **J2EE Connection**. The J2EE Transaction Breakdown dialog box opens.



Connection Status: Disconnected

2. Enter the name or IP address of the application server and the mediator. The mediator default is *localhost*.

3. Click **Connect**. The J2EE Diagnostics module monitor attempts to connect the mediator and the application server.

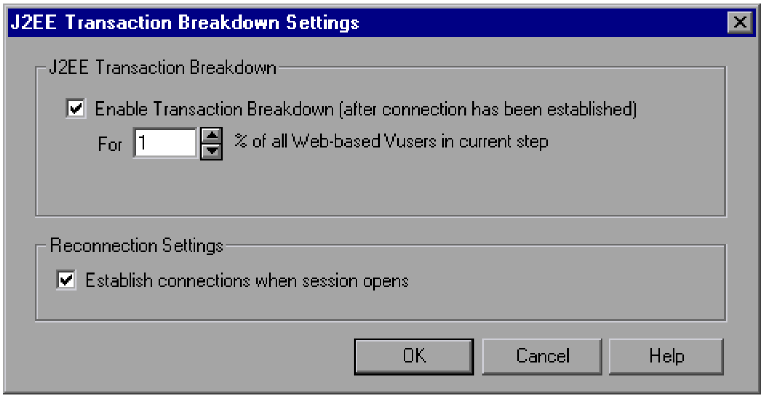
4. When the connection is successful, click **Close**.

   If the connection is not successful, see the Output window for more information. You can view the Output window from the J2EE Diagnostics tab by clicking the **Errors** link in the status bar.

Note: Only Vusers that start running after the connection has been established will participate in the breakdown.

To enable J2EE Diagnostics on the Console:

1 Choose **J2EE** > **Settings**. The J2EE Transaction Breakdown Settings dialog box opens.



J2EE Transaction Breakdown Settings

J2EE Transaction Breakdown

☑ Enable Transaction Breakdown (after connection has been established)

For 1 % of all Web-based Vusers in current step

Reconnection Settings

☑ Establish connections when session opens

2 To enable transaction breakdown monitoring, select **Enable Transaction Breakdown** and set the percent of users to participate in the monitoring. The percent determines how many of the transactions on the application server are reported to the Console. Reducing this number will reduce the overhead on the application server. The minimum is 1% or 1 virtual user, whichever is more. For example, if you enter a sampling value of 25% and run 12 Vusers in script1, 8 Vusers in script2, and 1 Vuser in script3, transaction breakdown will be performed on 3 Vusers in script1, 2 Vusers in script2, and 1 Vuser in script3.

Note: You can only enable transaction breakdown monitoring after the connection to the mediator and the application server has been established.

During a scenario or step run, you cannot change the percentage value.

3 To automatically establish a connection every time a session is opened, select **Establish connections when scenario/session opens**.

Viewing J2EE Diagnostics Data

This section describes the following topics:

- Viewing J2EE Diagnostics Data
- Showing J2EE Measurement Breakdown
- Example Transaction Breakdown
- J2EE Transaction Breakdown Diagnostics Graphs

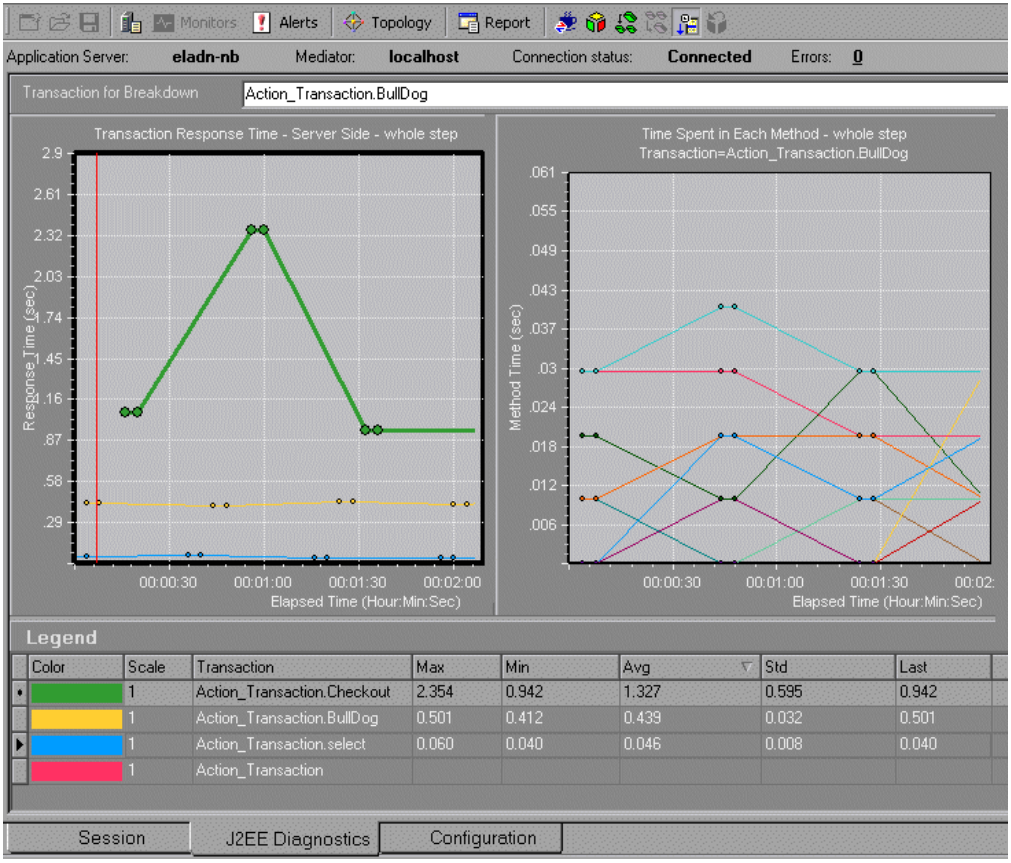
Viewing J2EE Diagnostics Data

You view transaction breakdown data using the J2EE Diagnostics tab in the Tuning Console.

To open the J2EE Diagnostics tab, perform one of the following:

- Select the **J2EE Diagnostics** tab.

➤ From the Session tab, right-click the Transaction Response Time graph and choose **Show J2EE server side**.



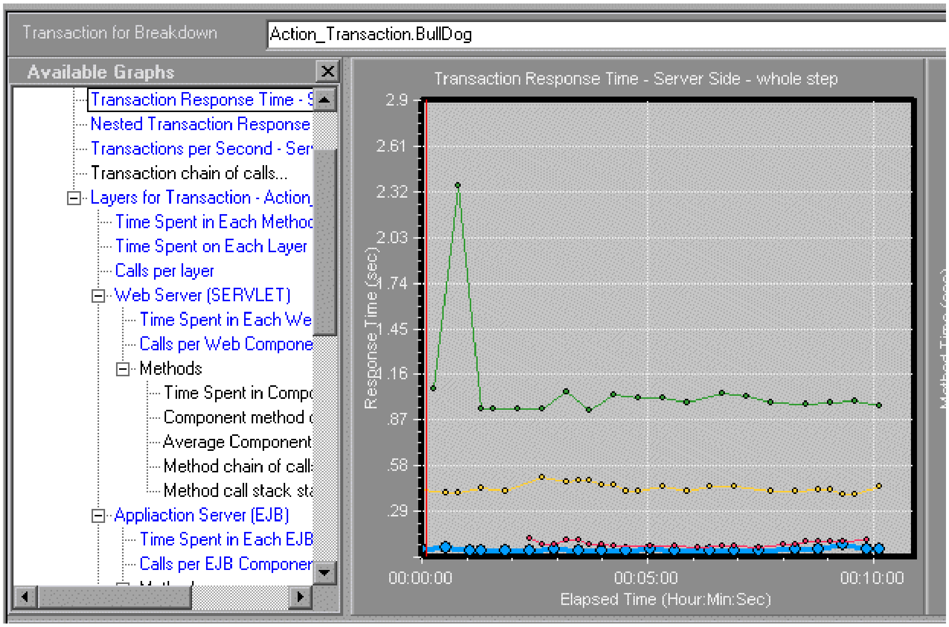
Application Server: **eladn-nb** Mediator: **localhost** Connection status: **Connected** Errors: **0**

Transaction for Breakdown: Action_Transaction.BullDog

Legend

Color	Scale	Transaction	Max	Min	Avg	Std	Last
🟩	1	Action_Transaction.Checkout	2.354	0.942	1.327	0.595	0.942
🟨	1	Action_Transaction.BullDog	0.501	0.412	0.439	0.032	0.501
🟦	1	Action_Transaction.select	0.060	0.040	0.046	0.008	0.040
🟥	1	Action_Transaction					

To display the list of available transaction breakdown graphs, right-click in the graph and choose **Show Available Graphs**. The list of available graphs is displayed with the current graph outlined.



Note: The J2EE Diagnostics Module graphs are only available if there is an active connection between the application server and the mediator.

Understanding the J2EE Diagnostics Tab View

The J2EE Diagnostics tab displays the following J2EE transaction breakdown information:

J2EE Transaction Breakdown graphs: Display the J2EE transaction breakdown information in graph format.

Connection Statistics/Status bar: Displays the name of the application server and the mediator, their connection status, and the number of errors in this step.

Transaction for breakdown: Enables you to choose a transaction for breakdown. You can use this to switch from any level of detail within one transaction to the same level of detail in a different transaction without requiring you to move up and down the entire breakdown chain.

Legend: The legend displays statistics on the data displayed in the graph. By default, the legend is sorted according to descending average response time. As this value is updated at every refresh period, the component order can change often, making it difficult to track components. To keep the components sorted in the current order, right-click in the legend area and choose **Keep the Legend Sorted**, or choose **J2EE** > **Keep Legend Sorted**.

Showing J2EE Measurement Breakdown

You can break down the transactions displayed in the *Transaction Response Time - whole scenario/step* graph. From the transaction level, you can break down successively to the layer, class, and method level.

To view transaction breakdown graphs:

1 On the Run tab or the J2EE Diagnostics tab, display the **Transaction Response Time - whole scenario/step** graph.

2 Right-click a transaction, and choose **Show J2EE server side**.

A graph showing the next level of breakdown is displayed in the J2EE Diagnostics tab.

➤ To view successive breakdown graphs, right-click in the relevant graph area and choose **Show Measurement Breakdown**, or choose **J2EE** > **Show Measurement Breakdown**. Each time you change the breakdown level, the selected graph is outlined in the Available Graphs list.

- ➤ To return to the previous breakdown level, right-click in the graph and choose **Undo Measurement Breakdown**, or choose **J2EE** > **Undo Measurement Breakdown**.

  For information on the available graphs, see "J2EE Transaction Breakdown Diagnostics Graphs" on page 329.

Note: For nested transactions, for example when transaction B is a subtransaction within transaction A, the subtransaction finishes earlier than the parent, and is therefore, reported to the Console earlier than the parent transaction. This may cause data that is displayed for a parent transaction and a subtransaction not to be vertically aligned on the graph.

Viewing Call Chains and Call Stack Statistics

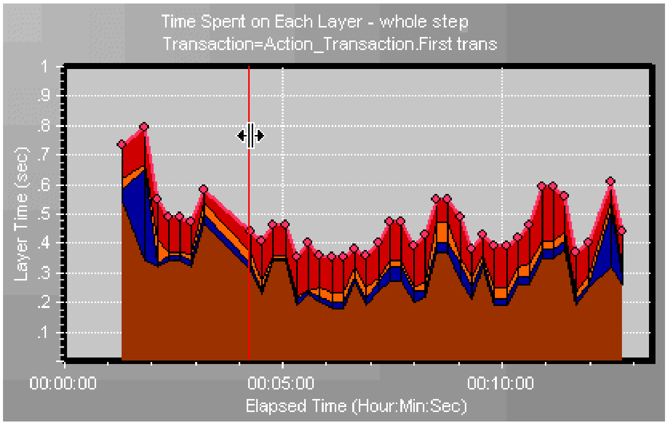
You can view the chain of calls for transactions and methods. The chain of calls answers the question "Who did I call?"

You can also view the call stack statistics for methods. Call stack statistics answer the question "Who called me?"

Chain of call and call stack statistics data are shown in the Calls window. The title of the window changes depending on which kind of data you are viewing.

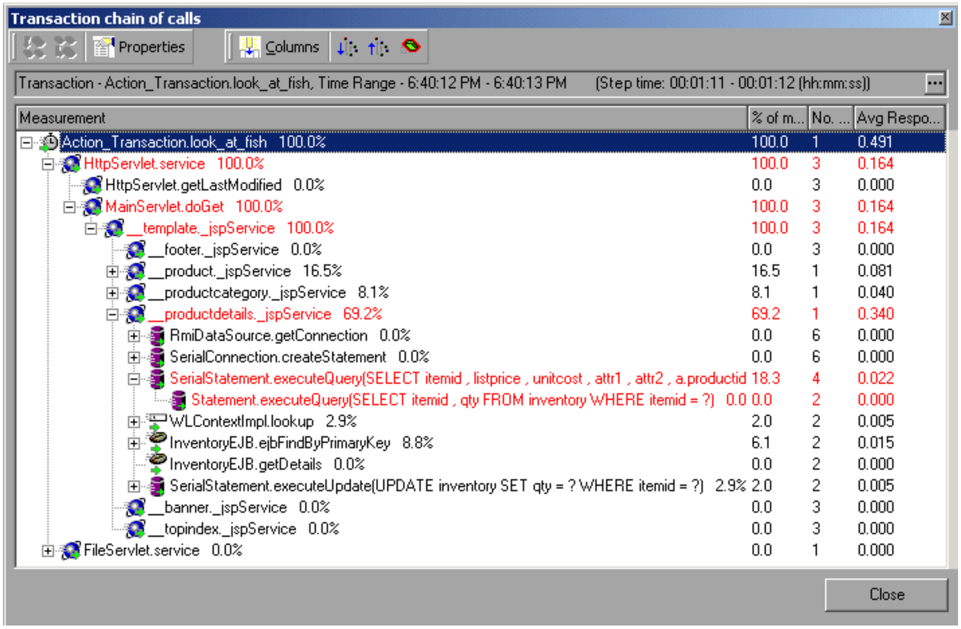
- ➤ To view call chains, right-click a component, and choose **Show <Component> Chain of Calls**, or choose **J2EE** > **Show chain of calls**. The Calls window opens displaying the components that this component called.
- ➤ To view stack statistics, right-click a method, and choose **Show Method Call Stack Statistics**, or choose **J2EE** > **Method call stack**. The Calls window opens displaying all the components that called the selected method.

To change the point to which the Calls window relates, drag the red time line to the desired spot.



Understanding the Call Windows

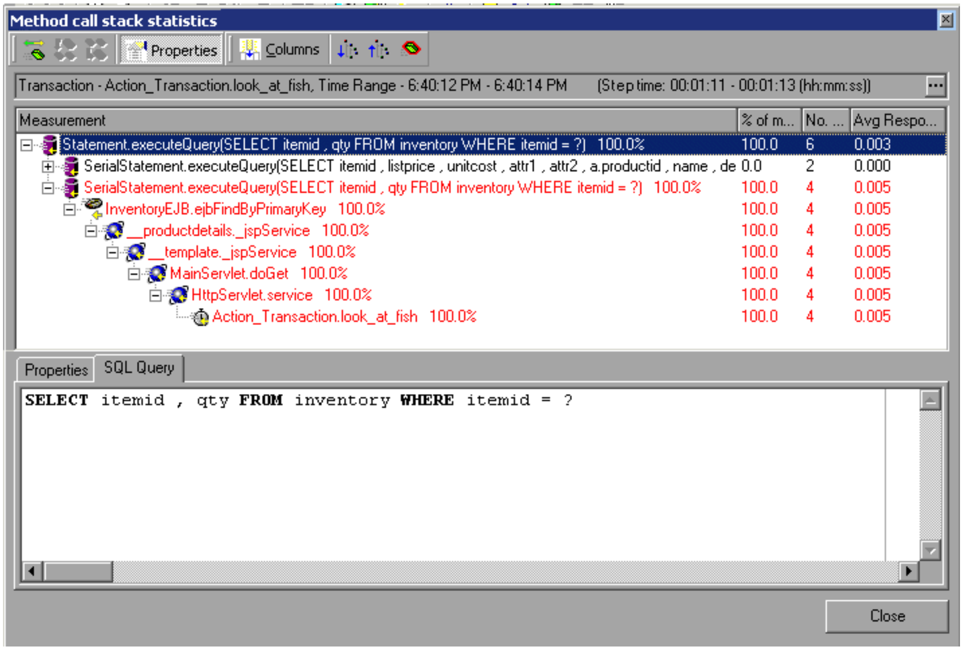
You use the Call Chain window to view the components that the selected transaction called. In the following figure, all the calls in the critical path of the *Action_Transaction.look_at_fish* transaction are displayed.



Transaction - Action_Transaction.look_at_fish, Time Range - 6:40:12 PM - 6:40:13 PM (Step time: 00:01:11 - 00:01:12 (hh:mm:ss))

Measurement	% of m...	No. ...	Avg Respo...
Action_Transaction.look_at_fish 100.0%	100.0	1	0.491
HttpServlet.service 100.0%	100.0	3	0.164
HttpServlet.getLastModified 0.0%	0.0	3	0.000
MainServlet.doGet 100.0%	100.0	3	0.164
__template._jspService 100.0%	100.0	3	0.164
__footer._jspService 0.0%	0.0	3	0.000
__product._jspService 16.5%	16.5	1	0.081
__productcategory._jspService 8.1%	8.1	1	0.040
__productdetails._jspService 69.2%	69.2	1	0.340
RmiDataSource.getConnection 0.0%	0.0	6	0.000
SerialConnection.createStatement 0.0%	0.0	6	0.000
SerialStatement.executeQuery(SELECT itemid , listprice , unitcost , attr1 , attr2 , a.productid	18.3	4	0.022
Statement.executeQuery(SELECT itemid , qty FROM inventory WHERE itemid = ?) 0.0	0.0	2	0.000
WLContextImpl.lookup 2.9%	2.0	2	0.005
InventoryEJB.ejbFindByPrimaryKey 8.8%	6.1	2	0.015
InventoryEJB.getDetails 0.0%	0.0	2	0.000
SerialStatement.executeUpdate(UPDATE inventory SET qty = ? WHERE itemid = ?) 2.9%	2.0	2	0.005
__banner._jspService 0.0%	0.0	3	0.000
__topindex._jspService 0.0%	0.0	3	0.000
FileServlet.service 0.0%	0.0	1	0.000

You use the Call Stack Statistics window to view which components called the selected component. In the following figure, the *executeQuery(SELECT itemid* query was called by *InventoryEJB...*, which was called by *__product.details._jspService*, and so on, down to the transaction at the bottom of the chain.



Method call stack statistics

Transaction - Action_Transaction.look_at_fish, Time Range - 6:40:12 PM - 6:40:14 PM (Step time: 00:01:11 - 00:01:13 (hh:mm:ss))

Measurement	% of m...	No. ...	Avg Respo...
Statement.executeQuery(SELECT itemid , qty FROM inventory WHERE itemid = ?) 100.0%	100.0	6	0.003
SerialStatement.executeQuery(SELECT itemid , listprice , unitcost , attr1 , attr2 , a.productid , name , de	0.0	2	0.000
SerialStatement.executeQuery(SELECT itemid , qty FROM inventory WHERE itemid = ?) 100.0%	100.0	4	0.005
InventoryEJB.ejbFindByPrimaryKey 100.0%	100.0	4	0.005
__productdetails._jspService 100.0%	100.0	4	0.005
__template._jspService 100.0%	100.0	4	0.005
MainServlet.doGet 100.0%	100.0	4	0.005
HttpServlet.service 100.0%	100.0	4	0.005
Action_Transaction.look_at_fish 100.0%	100.0	4	0.005

Properties | SQL Query

```
SELECT itemid , qty FROM inventory WHERE itemid = ?
```

Switch to Method Chain of Calls: When the Call Stack Statistics data is displayed, displays the method Chain of calls data (only if the root is a method).

Switch to Method Call Stack Statistics: When the method Chain of calls data is displayed, displays the method Call Stack Statistics data (only if the root is a method).

Show Method Chain of Calls: Displays the Chain of Calls window.

Show Method Call Stack Statistics: Displays the Call Stack Statistics window.

Properties: Hides or displays the properties area (lower pane).

Columns: Enables you to select the columns shown in the Calls window. To display additional fields, drag them to the desired location in the Calls window. To remove fields, drag them from the Calls window back to the Columns chooser.

> **Measurement**: Name of the method, displayed as *classname.methodname*. In the case of a database call, query information is also displayed. The percent shown indicates the percent of calls to this component from its parent.
>
> **% of Transaction (or Root Method)**: Percentage of the total time of the transaction (or method) from the total time of the root tree item.
>
> **No. of Calls**: Displays the amount of times that this transaction or method was executed.
>
> **Avg Response Time**: Response time is the time from the beginning of execution until the end. Average response time is the total response time divided by the number of hits.
>
> **Min Response Time**: The minimum response time.
>
> **Max Response Time**: The maximum response time.
>
> **% of Caller**: Displays the execution time, as a portion of the time of the calling component.
>
> **Total time**: Displays the total method execution time, including the child execution time.

Expand Tree: Expands the entire tree.

Contract Tree: Contracts the entire tree.

Expand Worst Path: Expands only the parts of the path on the critical path.

Method Properties Area: Displays the full properties of the selected method.

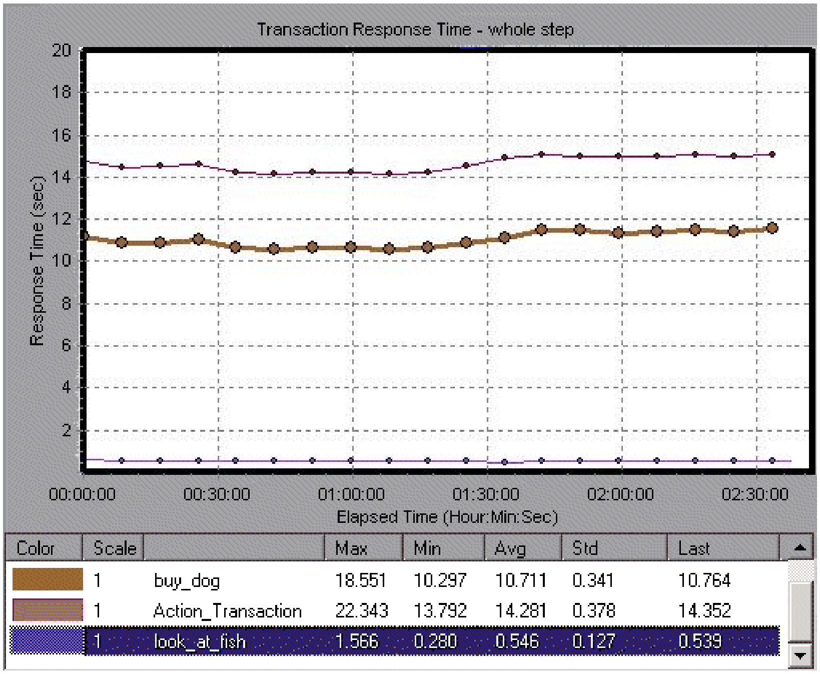
SQL Query: Displays the SQL query for the selected method. (For Database only.)

Example Transaction Breakdown

The following graphs illustrate the breakdown of a transaction to its layer, class, and methods.

Transaction Level

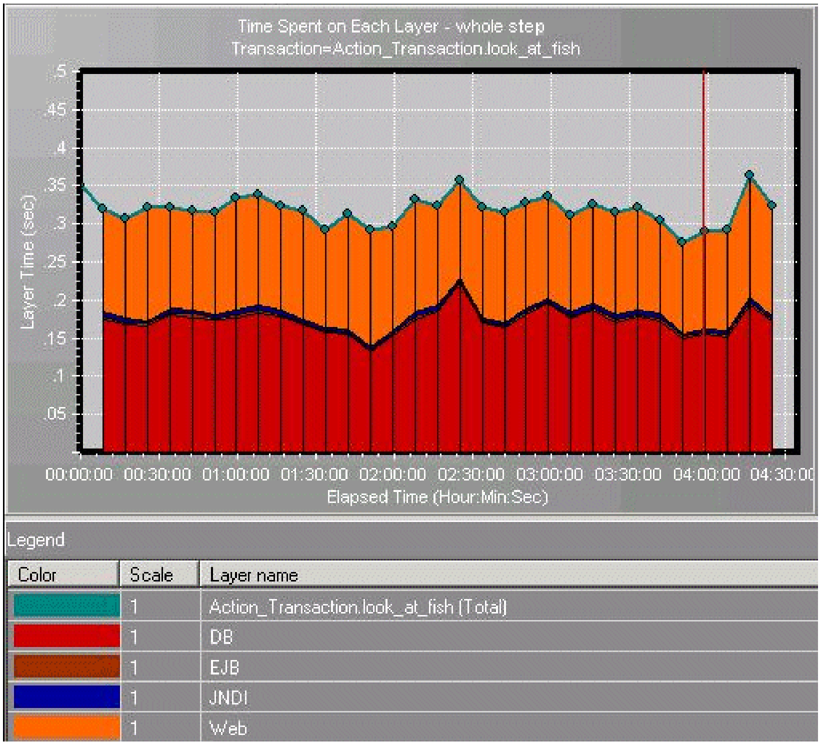
The following figure displays the top level Transaction Response Time graph. The graph displays three transactions.



Color	Scale		Max	Min	Avg	Std	Last
▇	1	buy_dog	18.551	10.297	10.711	0.341	10.764
▇	1	Action_Transaction	22.343	13.792	14.281	0.378	14.352
▇	1	look_at_fish	1.566	0.280	0.546	0.127	0.539

Layer Level

The following figure displays the breakdown of the *Action_Transaction.look_at_fish* transaction into its layers.

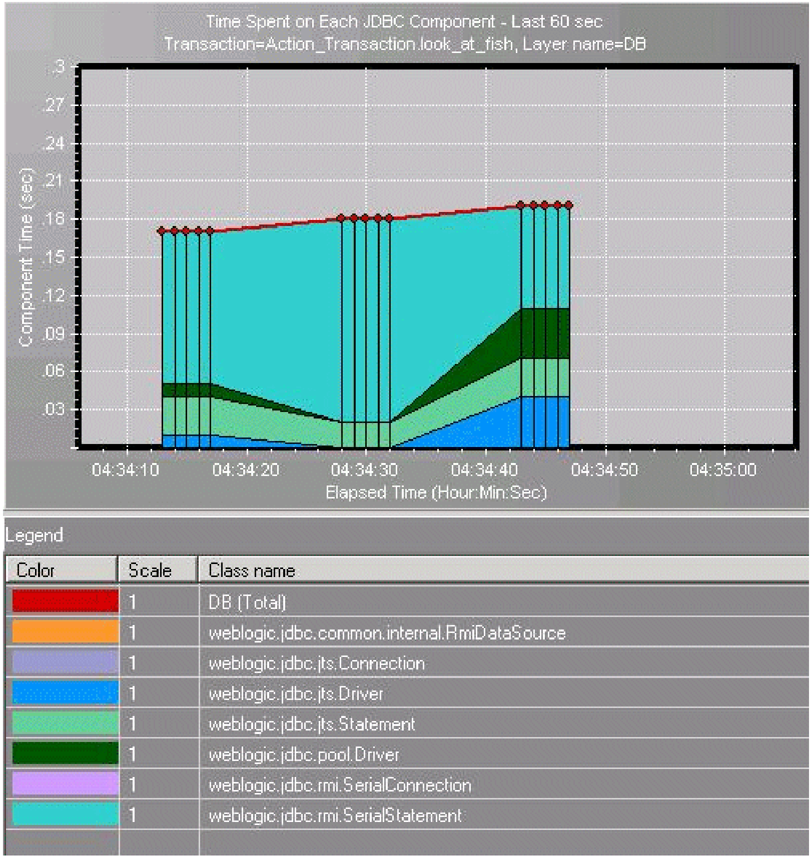


Color	Scale	Layer name
🟩	1	Action_Transaction.look_at_fish (Total)
🟥	1	DB
🟫	1	EJB
🟦	1	JNDI
🟧	1	Web

The top of each breakdown graph displays the name of the graph and the name of the parent component. In this *Time Spent on Each Layer* graph, the *Action_Transaction.look_at_fish* transaction has been broken down to its layers (DB, EJB, JNDI, and Web). In Web transactions, the database (DB) layer is generally the largest.

Class Level

The following figure displays the breakdown of the DB layer into classes.


Time Spent on Each JDBC Component - Last 60 sec
Transaction=Action_Transaction.look_at_fish, Layer name=DB
Component Time (sec)
.3
.27
.24
.21
.18
.15
.12
.09
.06
.03
04:34:10
04:34:20
04:34:30
04:34:40
04:34:50
04:35:00
Elapsed Time (Hour:Min:Sec)

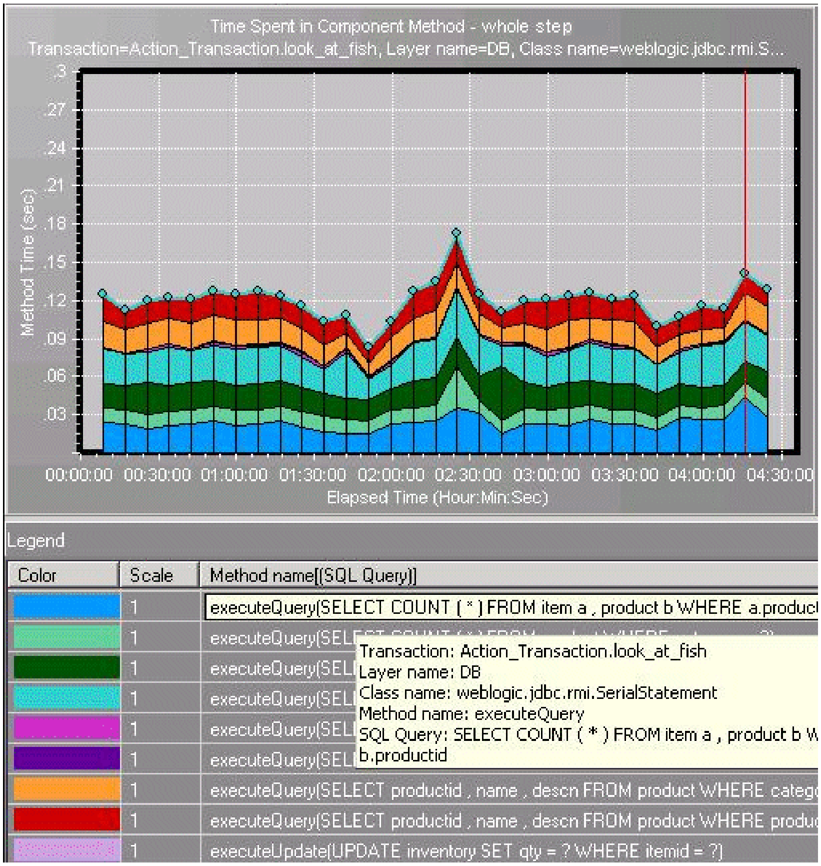
Legend

Color	Scale	Class name
🟥	1	DB (Total)
🟧	1	weblogic.jdbc.common.internal.RmiDataSource
🟪	1	weblogic.jdbc.jts.Connection
🟦	1	weblogic.jdbc.jts.Driver
🟩	1	weblogic.jdbc.jts.Statement
🟩	1	weblogic.jdbc.pool.Driver
🟪	1	weblogic.jdbc.rmi.SerialConnection
🟦	1	weblogic.jdbc.rmi.SerialStatement

In this *Time Spent on Each JDBC Component* graph, the DB layer of the *Action_Transaction.look_at_fish* transaction has been broken down to its classes. The Legend displays data on the classes.

Method/Query Level

The following figure displays the breakdown of a class into its methods:


Time Spent in Component Method - whole step
Transaction=Action_Transaction.look_at_fish, Layer name=DB, Class name=weblogic.jdbc.rmi.S...
Method Time (sec)
.3
.27
.24
.21
.18
.15
.12
.09
.06
.03
00:00:00
00:30:00
01:00:00
01:30:00
02:00:00
02:30:00
03:00:00
03:30:00
04:00:00
04:30:00
Elapsed Time (Hour:Min:Sec)
Legend

Color	Scale	Method name[(SQL Query)]
🟦	1	executeQuery(SELECT COUNT (*) FROM item a , product b WHERE a.product
🟩	1	executeQuery(SEL
🟩	1	executeQuery(SEL
🟦	1	executeQuery(SEL
🟪	1	executeQuery(SEL
🟪	1	executeQuery(SEL
🟧	1	executeQuery(SELECT productid , name , descn FROM product WHERE categ
🟥	1	executeQuery(SELECT productid , name , descn FROM product WHERE produ
🟪	1	executeUpdate(UPDATE inventory SET qty = ? WHERE itemid = ?)

Transaction: Action_Transaction.look_at_fish
Layer name: DB
Class name: weblogic.jdbc.rmi.SerialStatement
Method name: executeQuery
SQL Query: SELECT COUNT (*) FROM item a , product b W
b.productid

In this *Time Spent in Component Method* graph, the *SerialStatement* class of the DB layer of the *Action_Transaction.look_at_fish* transaction has been broken down to its methods. The Legend displays numerous method lines, differentiated by their SQL statements.

When you point to a method line, its properties are displayed in a pop-up window.

J2EE Transaction Breakdown Diagnostics Graphs

You can view J2EE transaction breakdown data in the following graphs:

Transaction Graphs

- **Transaction Response Time - Server Side**: Shows the transaction server response time of transactions that include steps that cause activity on the J2EE backend. The reported times, measured from the point when the transaction reached the Web server to the point it left the Web server, include only the time that was spent in the J2EE backend.
- **Nested Transaction Response Time - Server Side**: Shows all the transactions within the selected transaction, including the selected transaction.
- **Transactions per Second - Server Side**: Shows the amount of times per second that the transaction was executed.

Transaction Layer Graphs

- **Time Spent In Each Method**: Shows the time spent in each monitored method.
- **Time Spent On Each Layer**: Shows, for a specific transaction, the total time for each layer.
- **Calls per layer**: Shows, for a specific transaction how many times each layer was called per second.

Class Graphs

The class graphs can be displayed for Web server (SERVLET), application server (EJB), database (JDBC), or naming server (JNDI) classes:

- **Time Spent in Each <Class> Component**: Shows, for a specific transaction the total class time in the different components.
- **Calls per <Class> Component**: Shows how many times each component was called per second.

Method Graphs

The method graphs can be displayed for Web server (SERVLET), application server (EJB), database (JDBC), or naming server (JNDI) methods.

- **Time Spent In Each Component Method**: Shows the breakdown of the total layer time into the different methods/queries in a specific transaction.
- **Component Method Calls per Sec**: Shows how many times each method was called per second in a specific transaction.
- **Average Component Method Response Time**: Shows the average time spent in the method.

Examples of Modifying Application Server Configurations

This section provides examples modifying the configuration of the following application servers:

- Configuring the WebSphere Application Server
- Configuring the WebLogic Application Server
- Configuring the Oracle9i Application Server
- Configuring the JBoss Application Server

Configuring the WebSphere Application Server

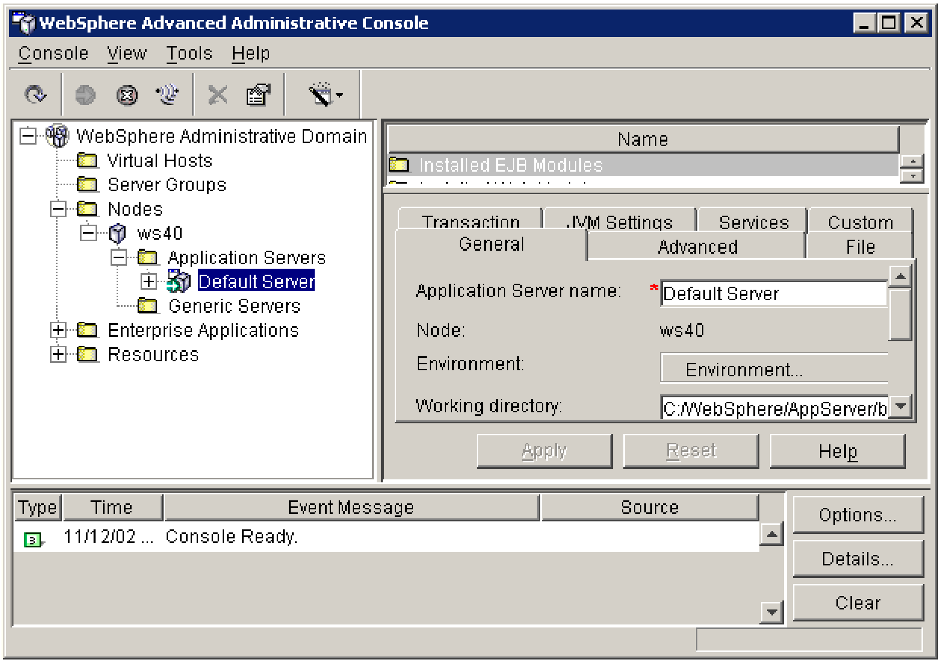
WebSphere servers are controlled by an Administrative Console. The Console has control over the JVM command line and allows you to add classpath elements, define runtime variables (**-D** variables), and configure the bootclasspath for WebSphere 4.x versions. You may also add any additional arguments to the JVM command line that may be needed.

Note that the Console for each WebSphere version may be different from the other versions. The examples shown in this section may not correspond exactly to your WebSphere version, but the principle is the same, that is, the parameters must be inserted in the appropriate fields.

Note: You must click **Apply** in each tab in the Administrative Console before moving to another tab in order to make the changes effective.

To configure the probe on a WebSphere application server:

1 Access the WebSphere Administrative Console, and display the General tab for the application server that you want to configure.



Application Server name: * Default Server

Node: ws40

Environment:

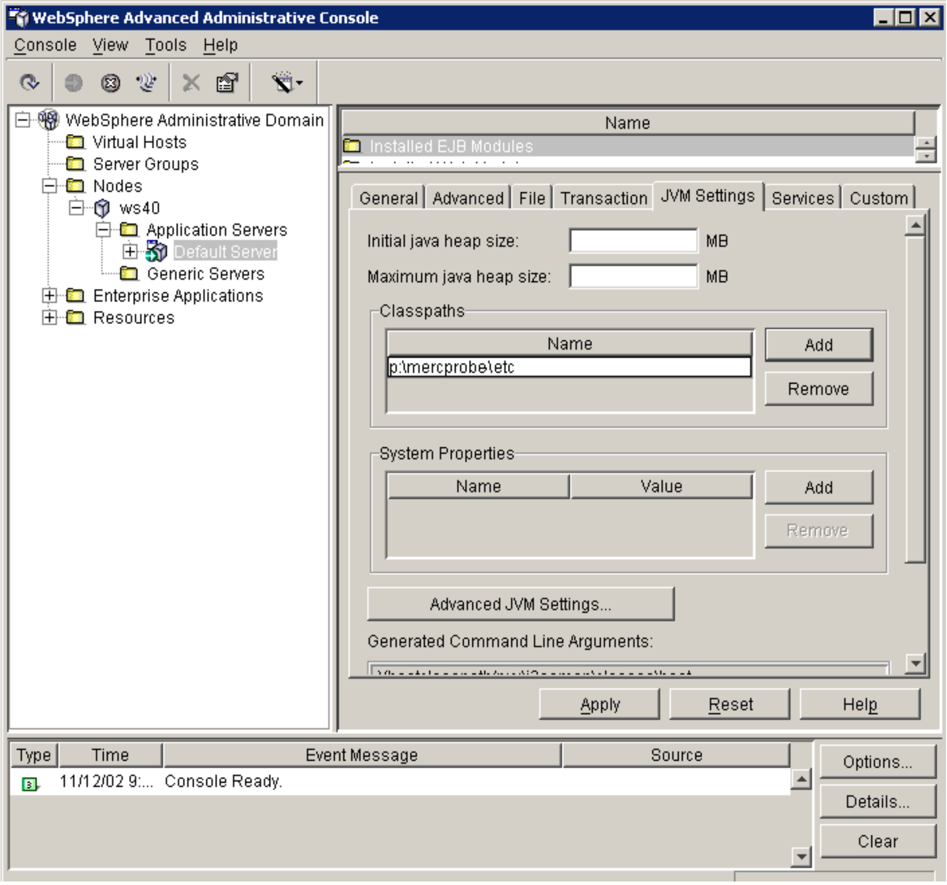
Working directory: C:/WebSphere/AppServer/b

Type	Time	Event Message	Source
B	11/12/02 ...	Console Ready.	

Note: This screen shows WebSphere version 4. Your console may appear different if you are running another version.

---

2 Click the **JVM Settings** tab.



Initial java heap size: MB

Maximum java heap size: MB

Name
p:\mercprobe\etc

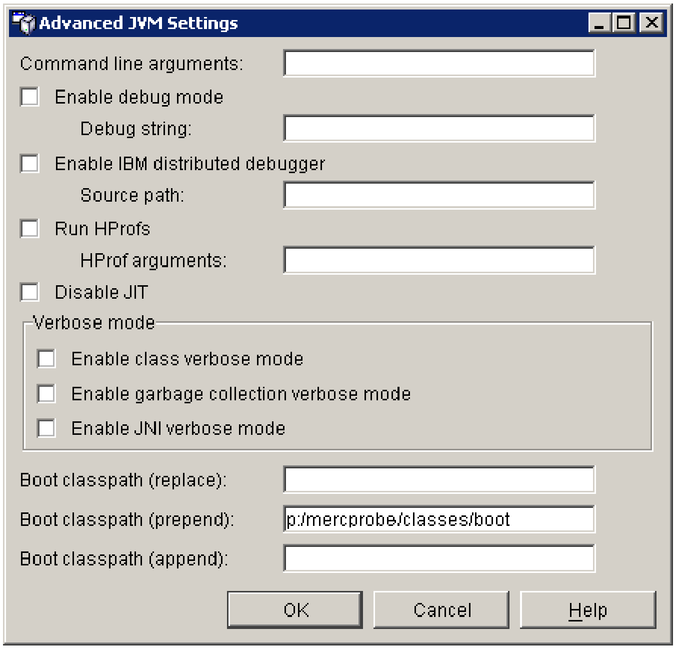
Generated Command Line Arguments:

Type	Time	Event Message	Source
	11/12/02 9:...	Console Ready.	

Note: If your JVM uses a JIT option, such as -hotspot or -classic, make sure (in the Java Command Line arguments on the console) that the -Xbootclasspath option is entered following the JIT option.

3 Click **Advanced JVM Settings**, to open the Advanced JVM Settings box.

4 Add the boot classpath elements to the prepend field, as in the following example.



Advanced JVM Settings

Command line arguments:

□ Enable debug mode

Debug string:

□ Enable IBM distributed debugger

Source path:

□ Run HProfs

HProf arguments:

□ Disable JIT

Verbose mode

□ Enable class verbose mode

□ Enable garbage collection verbose mode

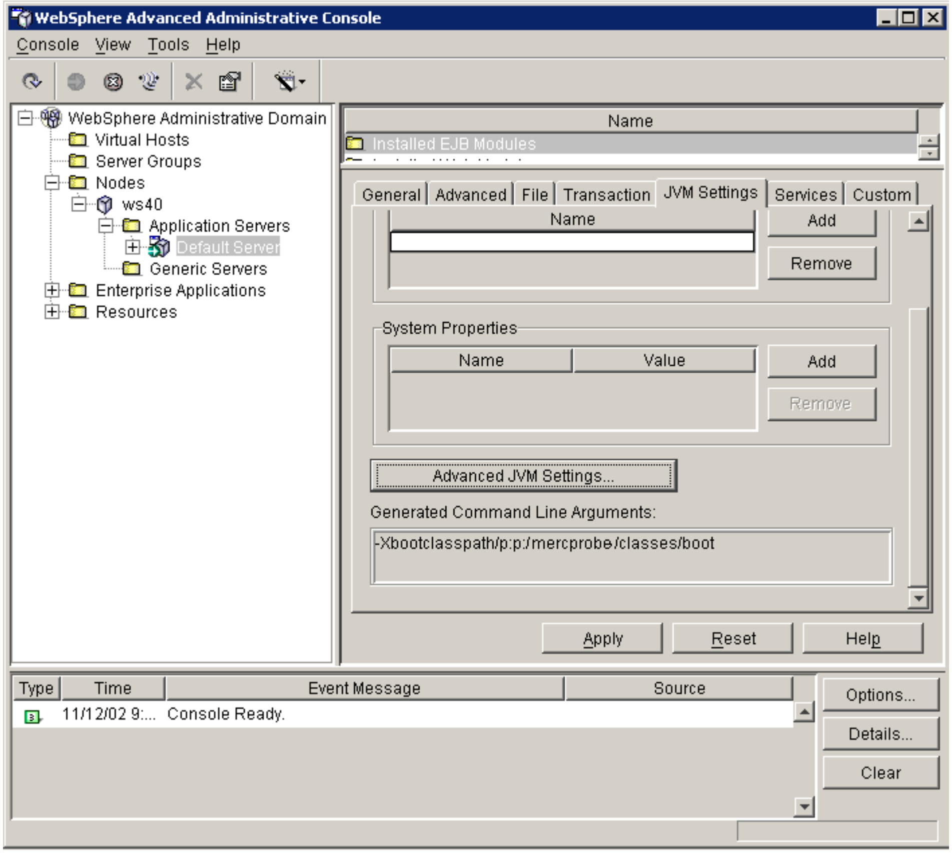
□ Enable JNI verbose mode

Boot classpath (replace):

Boot classpath (prepend): p:/mercprobe/classes/boot

Boot classpath (append):

5 In the Generated Command Line Arguments area, verify that the settings are the same as in the following example.



Generated Command Line Arguments:

```
-Xbootclasspath/p:p:/mercprobe/classes/boot
```

Type	Time	Event Message	Source
B	11/12/02 9:...	Console Ready.	

6 Restart the application server.

Configuring the WebLogic Application Server

These procedures explain how to configure the WebLogic application server so that it recognizes the J2EE probe, when WebLogic is running either as a script or a service.

To configure the WebLogic server, running as a script:

WebLogic application servers are started by running **shell** scripts (UNIX) or **cmd** scripts (Windows). While the standard installation comes with default scripts, it is very common for a site manager to heavily modify these scripts. For this reason, detailed instructions can not be given, and these guidelines should be used instead:

The J2EE probe is controlled through several command line arguments and, possibly, a few environment variables. The command line argument that might be used is **-Xbootclasspath**. You must add these command line arguments in a place that is appropriate to the specific installation. This could be in the actual file that launches the Java virtual machine, but may also be in a file that sets up a generic environment to be used by several WebLogic installed bases. It is important to understand the structure of the scripts that actually launch the application server when selecting the proper place to apply these changes.

You must take extra care when modifying environment variables since these may be changed by other parts of the script architecture after being set to reflect the probe installation.

Typical Locations of WebLogic Scripts

The following examples are provided to show some typical locations of the startup scripts for a Windows installation of WebLogic 6.1 and 7 along with the changes that need to be made to each script:

Weblogic 6.1

Script location example:
D:\bea\wlserver6.1\config\petstore\startPetStore.cmd.

Modify the Java command-line, and add the
-Xbootclasspath/p:<mercprobe>/classes/boot argument.

Weblogic 7

Script location example: **D:\bea\user_projects\petstore\startPetStore.cmd**.

To change the script, do the following immediately before the call to **startWLS.cmd** (at the end of the file):

➤ Add the following:

```
set JAVA_OPTIONS=-Xbootclasspath/p:d:\bea\user_projects\
petstore\mercprobe\classes\boot ;%JAVA_OPTIONS%
```

To configure the WebLogic server versions 7.x, 6.x or 5.x SP10 or later, when running as a service:

1. Locate the command file that is being used by the customer to install the WebLogic service (for example, **installNtService.cmd**).
2. Backup this file.

Tip: It is recommended that you check that the ***.cmd** file you found installs the service successfully, before you modify it. To verify that the ***.cmd** file installs the service successfully, run the file before making any changes to it.

---

3. On the line that begins with set CMDLINE=, add:

```
-Xbootclasspath/p:<mercprobe>/classes/boot
```

For example:

```
set CMDLINE=" -Xbootclasspath/p:D:/bea/Weblogic/
mercprobe/classes/boot -ms64m -mx64m -classpath..."
```

Note: If your JVM uses a JIT option, such as -hotspot or -classic, make sure that the -Xbootclasspath option is entered following the JIT option.

---

4 Save the file, and then run it. This installs the service.

WebLogic 8.1

To load the probe with the JRockit JVM:

1 Locate the command file that the user uses to invoke the WebLogic application server (for example, startWebLogic.cmd).

2 Create a new copy of the command file (for example, startWebLogicWithJRockit.cmd).

3 Check that the JAVA executable used with WebLogic is JRockit. To do this, check the following environment variable setting in startWebLogicWithJRockit.cmd (the newly-saved file]:

set JAVA_HOME=c:\bea\jrockit. For example, for WebLogic 8.1 it might be:

set JAVA_HOME=<BEA_HOME_DIR>\jrockit81sp1_141_03]

4 Check that JAVAPROBE_HOME is set. For example:

set JAVAPROBE_HOME=c:\JavaProbe

5 Find the JAVA invocation that starts the WebLogic application server. It should look something like this:

%JAVA_HOME%\bin\java %JAVA_VM% %JAVA_OPTIONS% .....
weblogic.Server

6 Add the following lines to the application server startup command line after %JAVA_OPTIONS%:

```
    -Xmanagement:class=com.mercury.opal.capture.proxy.JRockitManagement
    -Xbootclasspath/p:%JAVAPROBE_HOME%/classes/boot
```

Configuring the Oracle9i Application Server

This procedure explains how to configure the Oracle9i application server so that it recognizes the J2EE probe.

To configure the Oracle9i Application Server:

1 Configure the probe through the Oracle9i administration console (the Enterprise Manager):

- Access the URL **http://<application server>:<adminport>.**
- Click **Instance**.
- Click **Application**.
- At the bottom of the page, click **Server Properties**.
- Enter the location of the Java executable that was specified during installation of the probe.
- Add the following line to the Java options:

```
-Xbootclasspath/p:<mercprobe>/classes/boot
```

- Click **Apply**.
- Use the navigation path at the top of the page to return to the application server, and start or restart the application.
- Verify that the Application Server instance starts correctly in the console. Log files for the Application Server are typically located under <oracle-home>/opmn/logs/<application><instance>.

If the Application Server does not start properly, review the changes made in the previous steps.

Configuring the JBoss Application Server

This section explains how to configure the JBoss 3.2.1 application server.

To configure the JBoss 3.2.1 server, running as a script:

JBoss application servers are started by running **shell** scripts (UNIX) or **cmd** scripts (Windows). While the standard installation comes with default scripts, it is very common for a site manager to modify these scripts heavily. For this reason, detailed instructions cannot be given, and these guidelines should be used instead:

The J2EE probe is controlled through several command line arguments and, possibly, a few environment variables. The command line argument that might be used is **-Xbootclasspath** and **-classpath**.

You must add these command line arguments in a place that is appropriate to the specific installation. This could be in the actual file that launches the Java virtual machine, but may also be in a file that sets up a generic environment to be used by several JBoss installed bases. It is important to understand the structure of the scripts that actually launch the application server when selecting the proper place to apply these changes.

You must take extra care when modifying environment variables since these may be changed by other parts of the script architecture after being set to reflect the probe installation.

Typical Location of JBoss Scripts

Here is an example of a startup script location on a Windows installation of the JBoss application server, and the needed changes in the script: Script location example: **D:\JBoss\EJBContainer\bin\run.bat**.

Important: <mercprobe>/etc must be added to the beginning of the line.

1. Copy <Jboss Home>\bin\run.bat (or .sh) to <Jboss Home>\runMercury.bat (or .sh).
2. Open the new file runMercury.bat (or .sh).

3. Just before the Java command line used to start the server, add the following variables:

   For Windows platforms:

   ```
   set merc_probe=<J2EE Transaction Breakdown Home Directory>
   ```

   For UNIX platforms:

   ```
   merc_probe=<J2EE Transaction Breakdown Home Directory>
   ```

4. In the same section of the file, add a parameter to the command line:

   ```
   -Xbootclasspath/p:%merc_probe%\classes\boot
   ```

   Example (Windows):

   ```
   "%JAVA%" -Xbootclasspath/p:%merc_probe%\classes\boot
   %JAVA_OPTS% -classpath "%CLASSPATH%" org.jboss.Main %ARGS%
   ```

5. To start the application server with the probe, run runMercury.bat.

Troubleshooting the J2EE Diagnostics Module Monitor

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 Transaction Breakdown 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 Transaction Breakdown Monitor Initializer again.

Error Message Displayed When Running the Mediator on Localhost

When running the mediator on *localhost*, the following error may be displayed in the Console output window:

"Error: Communication error: An error occurred while calling the VirtualAlloc function. (sys error message - Attempt to access invalid address.)[MsgId: MERR-10343]"

This is caused by a high number of Vusers relative to the available resources of the machine on which they are running. To resolve this, run the mediator on a more powerful machine, reduce the number of Vusers, or reduce the percentage of Vusers being monitored.

No Data in the J2EE Diagnostics Module Monitor Graphs

If there is no data displayed in the J2EE Diagnostics Module monitor graphs, check the following:

- Ensure that you have installed all the J2EE Transaction Breakdown Diagnostics Module monitor components and that the probe is properly configured. For more information, see “Configuring the J2EE Diagnostics Module Monitor on the Application Server” on page 311.
- Ensure that the connection between the Console and the probe is active. For more information, see “Examples of Modifying Application Server Configurations” on page 330.
- Ensure that you selected a transaction in the Transaction Response Time - whole scenario/step graph. If a transaction is not selected, the J2EE graph will be blank.
- In some cases, a single iteration of a script may not provide enough data to be displayed. Increase the number of iterations.

Only One Transaction is Being Displayed in the Graphs

If only one transaction is being displayed in the transaction breakdown graphs, ensure that you have not used the same transaction name within multiple scripts in the same scenario or step.

The most common example of this is two scripts that define no inner transactions.

In this case, only one transaction, the *Action_transaction* will appear in the transaction breakdown graphs, and measurement data will be combined for both transactions.

Part VI

Monitoring a Session

27

Online Monitoring

You can monitor session step execution using the online monitors.

This chapter describes the online monitor user interface. The specific monitors are discussed in the *LoadRunner Monitor Reference*.

About Online Monitoring

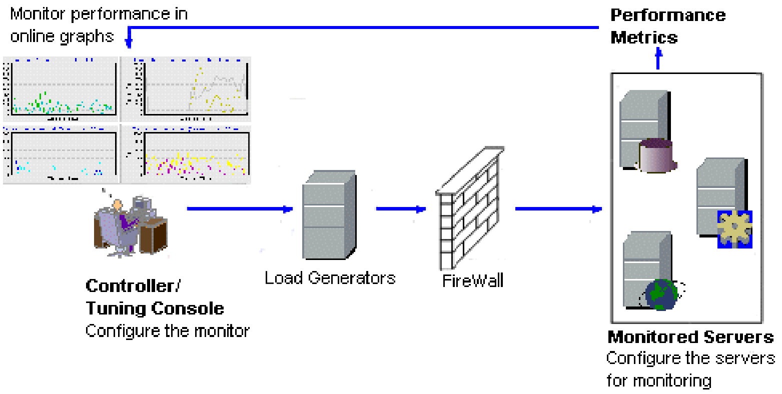
You can view data generated during session step execution using the online monitors. You specify the machines that the Console will monitor during a session step execution, and view the data collected by the monitors using the online graphs.

A primary factor in a transaction's response time is its resource usage. By monitoring resources during a session step run, you can determine why a bottleneck occurred on a particular machine. The server resource monitors let you keep track of resources used during a session step. The Console displays the selected resource monitors in real-time during test execution.

Note that you can select the server resource measurements to monitor both before and during the session step.

Setting Up the Monitoring Environment

Before monitoring a session step, you need to set up and configure the monitoring components. Each monitor has different configuration requirements which are explained in the specific monitoring chapters in the *LoadRunner Monitor Reference*. The diagram below shows the monitoring process.



Before monitoring a server, perform the following steps:

- configure the monitoring environment on the server machine (if necessary)
- configure the monitor on the Controller or Console machine

Configuring the Monitoring Environment on the Server Machine

To use the following monitors, you must first install or configure monitoring components on the server machine:

- COM+
- Citrix
- DB2
- IBM WebSphere MQ
- iPlanet (NAS)
- J2EE
- J2EE Transaction Breakdown
- .NET CLR
- Network Delay
- Oracle
- PeopleSoft (Tuxedo)
- SAP
- SAPGUI
- SAP Portal
- SAP CCMS
- Siebel Server Manager
- Siebel Web Server
- SiteScope
- Sybase
- Tuxedo
- UNIX
- WebLogic (JMX)
- WebSphere (EPM)
- WebSphere

Configuring the Monitor on the Console Machine

To obtain performance data for a monitor, you need to enable the monitor (from the Console), and indicate which statistics and measurements you want to monitor. You select these counters using the monitor's Add Measurements dialog box.

For more information on setting up the monitoring environment and configuring a monitor, see the specific monitoring chapter in the *LoadRunner Monitor Reference*.

Monitor Types

The online monitors are divided into the following categories:

- Run-Time Monitors
  Display the number and status of Vusers participating in the session step, as well as the number and types of errors that the Vusers generate. For more information, see refer to the chapter “Run-Time and Transaction Monitoring” in the *LoadRunner Monitor Reference*.
- Transaction Monitors
  Display the transaction rate and response time during session step execution. For more information, see refer to the chapter “Run-Time and Transaction Monitoring” in the *LoadRunner Monitor Reference*.
- Web Resource Monitors
  Provide information about the number of Web connections, throughput volume, HTTP responses, server retries, and pages downloaded to the Web servers during the session step. For more information, refer to the chapter “Web Resource Monitoring” in the *LoadRunner Monitor Reference*.
- System Resource Monitors
  Measure the Windows, UNIX, Tuxedo, SNMP, Antara FlameThrower, and SiteScope resources used during a session step. For more information, refer to the chapter “System Resource Monitoring” in the *LoadRunner Monitor Reference*.
- Network Delay Monitor
  Displays information about the network delays on your system. For more information, refer to the chapter “Network Monitoring” in the *LoadRunner Monitor Reference*.
- Firewall Monitor
  Measures statistics of the firewall servers during the session step. For more information, refer to the chapter “Firewall Server Performance Monitoring” in the *LoadRunner Monitor Reference*.
- Web Server Resource Monitors
  Measure statistics of the Apache, Microsoft IIS, iPlanet (SNMP), and iPlanet/Netscape Web servers during the session step. For more information, refer to the chapter “Web Server Resource Monitoring” in the *LoadRunner Monitor Reference*.

➤ Web Application Server Resource Monitors
Measure statistics 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 the session step. For more information, refer to the chapter "Web Application Server Resource Monitoring" in the *LoadRunner Monitor Reference*.

➤ Database Server Resource Monitors
Measure statistics of the SQL server, Oracle, Sybase, and DB2 databases during the session step. For more information, refer to the chapter "Database Resource Monitoring" in the *LoadRunner Monitor Reference*.

➤ Streaming Media Monitors
Measure statistics of the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer client during the session step. For more information, refer to the chapter "Streaming Media Monitoring" in the *LoadRunner Monitor Reference*.

➤ ERP/CRM Server Resource Monitors
Measure statistics of the SAP R/3 system, SAP Portal, Siebel Server Manager, Siebel Web Server, and PeopleSoft (Tuxedo) servers during the session step. For more information, refer to the chapter "ERP/CRM Server Resource Monitoring" in the *LoadRunner Monitor Reference*.

➤ Java Performance Monitors
Measure statistics of Java 2 Platform, Enterprise Edition (J2EE) objects, and Java-based applications, using J2EE machines. For more information, refer to the chapter "J2EE Performance Monitoring" in the *LoadRunner Monitor Reference*.

➤ J2EE Transaction Breakdown Monitors
Provide information to trace, time, and troubleshoot individual transactions through J2EE Web, application, and database servers. For more information, refer to "J2EE Diagnostics Module" on page 307.

➤ Application Component Monitors
Measure statistics of the Microsoft COM+ and Microsoft .NET CLR servers during a session step run. For more information, refer to the chapter "Application Component Monitoring" in the *LoadRunner Monitor Reference*.

- Application Deployment Solutions Monitor
  Measures statistics of the Citrix MetaFrame XP and 1.8 servers during a session step run. For more information, refer to the chapter “Application Deployment Solutions” in the *LoadRunner Monitor Reference*.

- Middleware Performance Monitors
  Measure statistics of the Tuxedo and IBM WebSphere MQ servers during a session step run. For more information, refer to the chapter “Middleware Performance Monitoring” in the *LoadRunner Monitor Reference*.

- Application Traffic Management Monitor
  Measures statistics of the F5 BIG-IP server during a session step run. For more information, refer to the chapter “Application Traffic Management Monitoring” in the *LoadRunner Monitor Reference*.

- Security Monitor
  Displays information about simulated attacks on the server during a session step using the Distributed Denial of Service graph. For more information, refer to the chapter “Security Monitoring” in the *LoadRunner Monitor Reference*.

- Infrastructure Resources Monitor
  Measure statistics of the network client data points during a session step run. For more information, refer to the chapter “Infrastructure Resources Monitoring” in the *LoadRunner Monitor Reference*.

For information on configuring graph settings and measurements, and exporting graph data, see Chapter 28, “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: Application component monitors are available only in the LoadRunner Controller. Application traffic and security monitors are only available in the Mercury Tuning Console.

For a detailed list of LoadRunner's monitors, see Mercury's Web site (http://www.mercuryinteractive.com/products/loadrunner/load_testing_monitors/supported.html).

Choosing Monitors and Measurements on the Tuning Console

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 measurements on the database server.

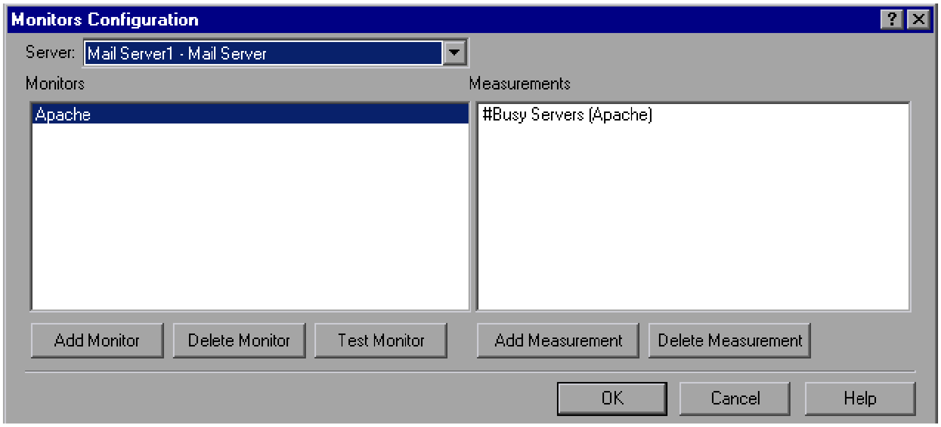
The relevant measurements will appear on the graphs belonging to the individual topology elements.

To select measurements to monitor via the System Topology window:

➤ Follow the directions in "Manually Selecting Monitors," on page 43.

To select measurements to monitor via the Monitors button on the main toolbar:

1 Click **Monitors**. The Monitors Configuration dialog box is displayed:



Monitors Configuration

Server: Mail Server1 - Mail Server

Monitors: Apache

Measurements: #Busy Servers (Apache)

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 opens. Choose the monitor and the measurements for the specific server.

Note: For detailed instructions on using the **Select Measurements to Monitor** dialog box, see “Manually Selecting Monitors,” on page 43.

---

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.

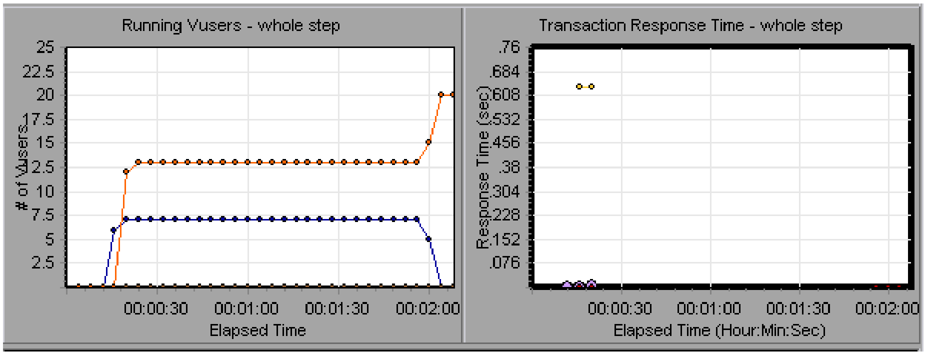
Viewing the Monitors in the Console

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 start 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.

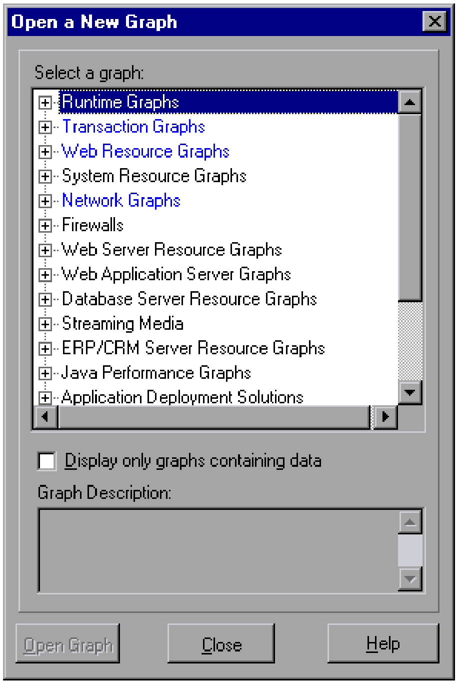
Opening Online Monitor Graphs in the Tuning Console

By default, Mercury Tuning Console displays four graphs in the Execute view: Running Vusers, Transaction Response Time, Hits per Second, and Windows Resources. 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.

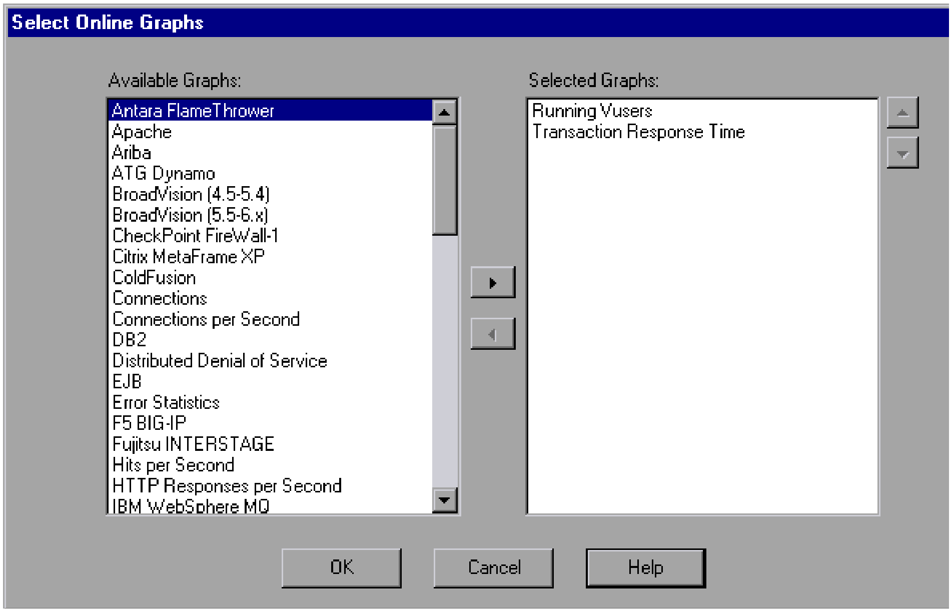


2 Click the "+" in the left pane to expand the graph tree, and select a graph. You can view a description of the graph in the **Graph Description** box. To view only graphs which contain data, select the **Display only graphs containing data** check box.

3 Click **Open Graph**. The graph appears in the graph view area.

To specify the graphs via the Select Online Graphs dialog box:

Click the **Graphs** button in the toolbar. The Select Online Graphs dialog box opens.



Available Graphs:

Antara FlameThrower
Apache
Ariba
ATG Dynamo
BroadVision (4.5-5.4)
BroadVision (5.5-6.x)
CheckPoint FireWall-1
Citrix MetaFrame XP
ColdFusion
Connections
Connections per Second
DB2
Distributed Denial of Service
EJB
Error Statistics
F5 BIG-IP
Fujitsu INTERSTAGE
Hits per Second
HTTP Responses per Second
IBM WebSphere MQ

Selected Graphs:

Running Vusers
Transaction Response Time

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.

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 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

You can 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.

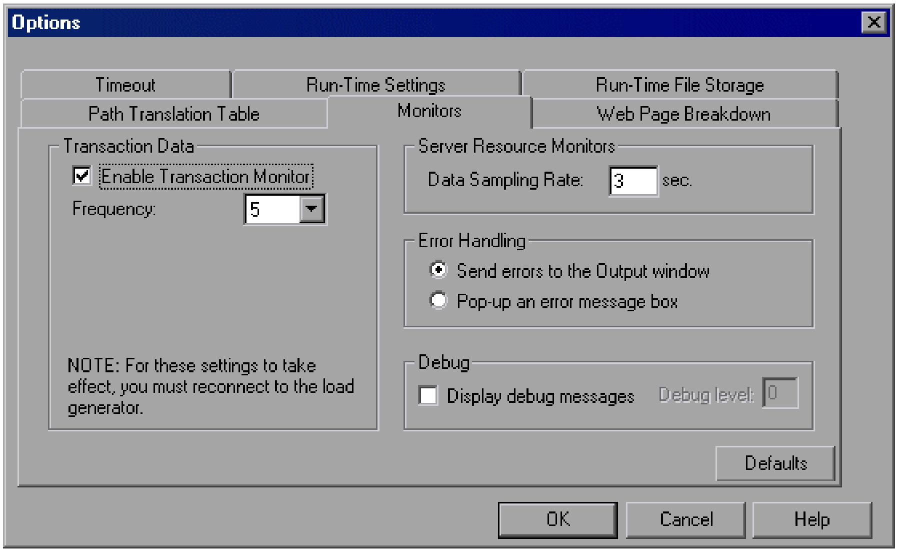
Setting Monitor Options

Before running your session step, LoadRunner 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.



Transaction Data

☑ Enable Transaction Monitor

Frequency: 5

NOTE: For these settings to take effect, you must reconnect to the load generator.

Server Resource Monitors

Data Sampling Rate: 3 sec.

Error Handling

● Send errors to the Output window

○ Pop-up an error message box

Debug

□ Display debug messages Debug level: 0

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, refer to the chapter "Run-Time and Transaction Monitoring" in the *LoadRunner Monitor Reference*.

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 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 "Configuring Graph Properties" on page 367.

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 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.

28

Configuring Online Graphs

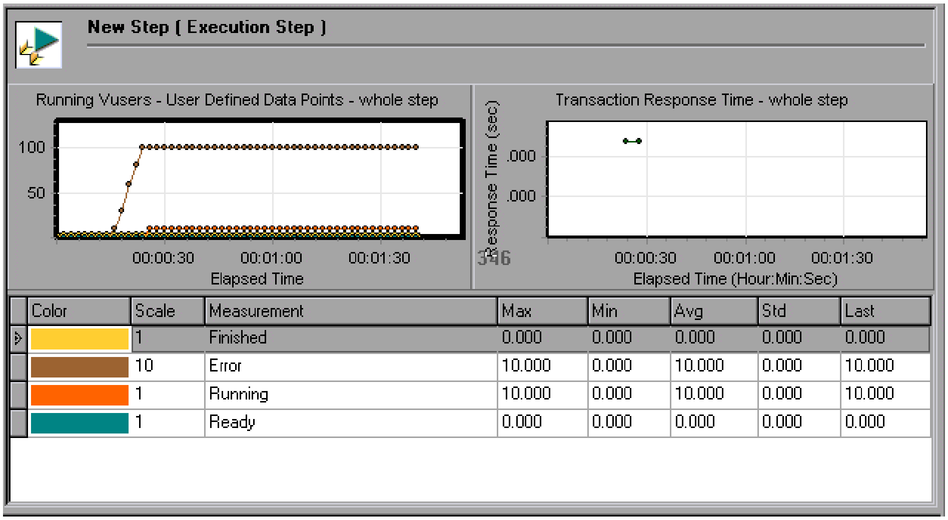
You can view the data collected by the monitors using the Mercury Tuning Console online monitor graphs.

- Configuring Graph Properties
- Configuring Graph Measurements
- Merging Graphs
- Adding Measurement Alerts
- Exporting Online Monitor Graphs
- Viewing Data Offline

About Online Monitor Graphs

Online monitor graphs display performance measurements for those resources being monitored during session step execution. Each measurement is represented on the graph by a colored line. Information about the measurements is listed in the legend below the graph. 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 in the Tuning Console,” on page 357, and “Setting Monitor Options” on page 360.


New Step (Execution Step)
Running Vusers - User Defined Data Points - whole step
100
50
00:00:30
00:01:00
00:01:30
Elapsed Time
Transaction Response Time - whole step
Response Time (sec)
.000
.000
346
00:00:30
00:01:00
00:01:30
Elapsed Time (Hour:Min:Sec)

	Color	Scale	Measurement	Max	Min	Avg	Std	Last
▷	🟨	1	Finished	0.000	0.000	0.000	0.000	0.000
	🟫	10	Error	10.000	0.000	10.000	0.000	10.000
	🟧	1	Running	10.000	0.000	10.000	0.000	10.000
	🟦	1	Ready	0.000	0.000	0.000	0.000	0.000

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

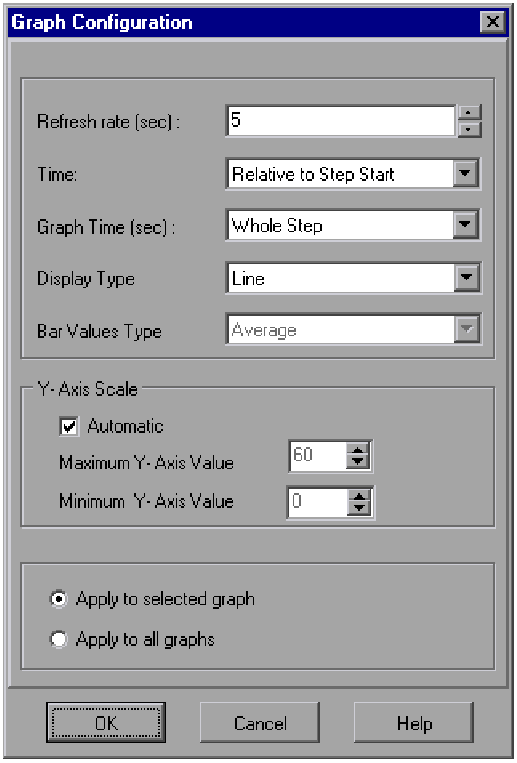
Mercury Tuning Console lets you configure the settings for your online monitor graphs. You can customize your graph in the following areas:

- Refresh Rate
- Time
- Graph Time
- Display Type
- Bar Values Type
- Y-Axis Scale
- 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 a graph and select **Configure**. The Graph Configuration dialog box opens.



Graph Configuration

Refresh rate (sec) : 5

Time: Relative to Step Start

Graph Time (sec) : Whole Step

Display Type: Line

Bar Values Type: Average

Y-Axis Scale

☑ Automatic

Maximum Y-Axis Value: 60

Minimum Y-Axis Value: 0

● Apply to selected graph

○ Apply to all graphs

2. Enter the desired refresh rate—the time between graph updates—in the Refresh Rate box.
3. Select a style for the x-axis from the Time box.
4. Select a value from the Graph Time box. The graph time is the time in seconds displayed by the x-axis.
5. Select a graph style from the Display Type box.
6. 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**.

7. Select a maximum or minimum value for the y-axis, or choose **Automatic** to view graphs using the default y-axis scale.
8. To apply the dialog box settings to all graphs, select **Apply to all graphs**.
9. 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.
10. 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

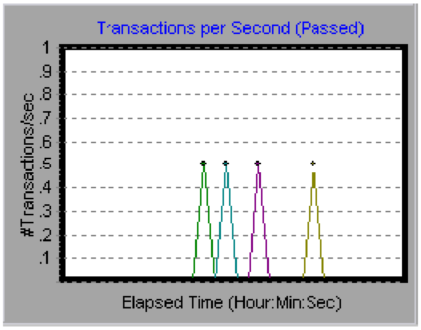
The interval in which the graph is refreshed with new data. By default, the graph is refreshed every five seconds. If you increase the refresh rate, the data is refreshed less frequently. Note that in a large load test, it is recommended to use a refresh rate of three to five seconds. This enables you to avoid problems with CPU resource usage.

Time

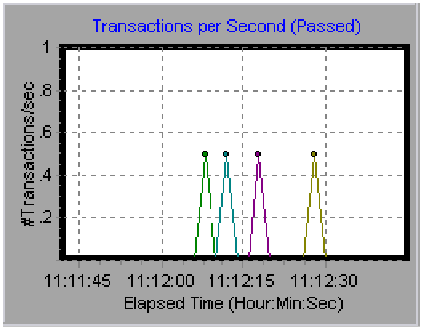
You can specify how the graph displays the x-axis time.

- **Don't Show:** Instructs Mercury Tuning Console 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 Mercury Tuning Console 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 Values Type

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 Scale

Instruct Mercury Tuning Console to display graphs using the default y-axis scale, or you specify a different y-axis scale. Click **Automatic** if you want Mercury Tuning Console 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:

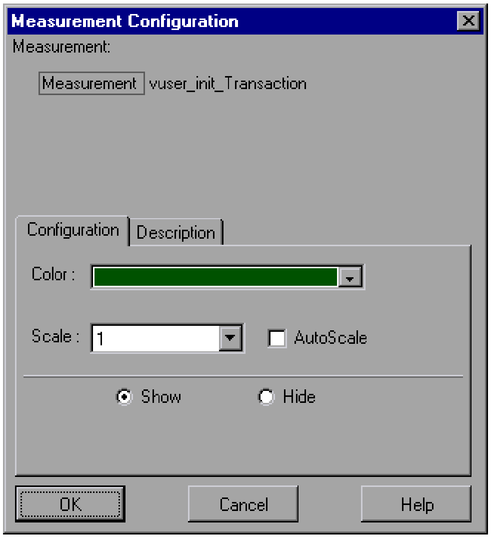
- Change Line Colors
- Set Measurement Scale
- Show and Hide Transactions

Changing Line Colors

Mercury Tuning Console 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.



Measurement Configuration

Measurement:

Measurement	vuser_init_Transaction

Color : 🟩

Scale : 1

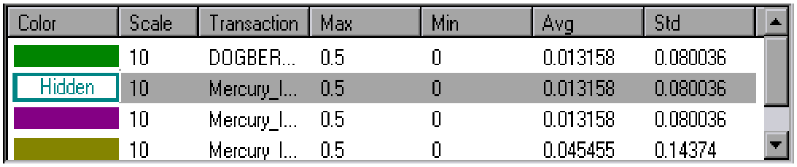
□ AutoScale

● Show ○ Hide

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.



Color	Scale	Transaction	Max	Min	Avg	Std
🟩	10	DOGBER...	0.5	0	0.013158	0.080036
Hidden	10	Mercury_l...	0.5	0	0.013158	0.080036
🟪	10	Mercury_l...	0.5	0	0.013158	0.080036
🟨	10	Mercury_l...	0.5	0	0.045455	0.14374

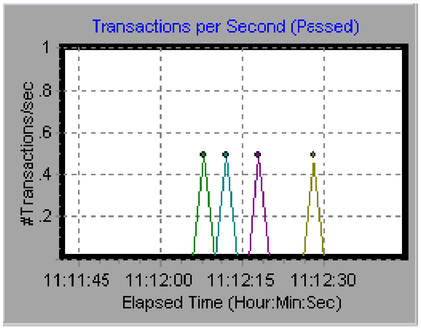
Setting the Scale of the Measurement

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.

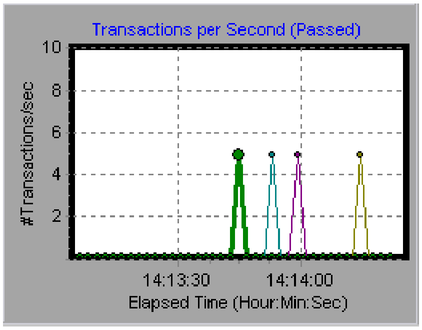
To set the scale of a measurement:

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.



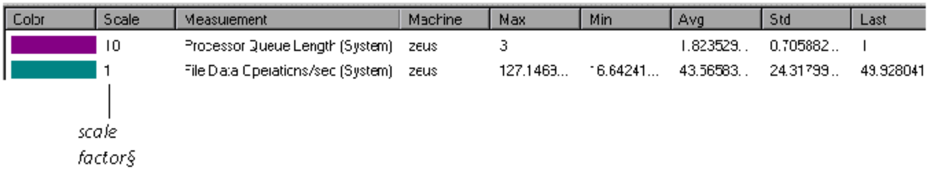
scale = 1



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.



Color	Scale	Measurement	Machine	Max	Min	Avg	Std	Last
🟪	10	Processor Queue Length (System)	zeus	3		1.823529..	0.705882..	1
🟦	1	File Data Operations/sec (System)	zeus	127.1463...	16.64241...	43.56583..	24.31799..	49.928041

By default, Mercury Tuning Console uses the *autoscale* option, which automatically scales the measurements by calculating the best ratio for displaying the graph.

Hiding and Showing 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 to focus on a specific transaction.

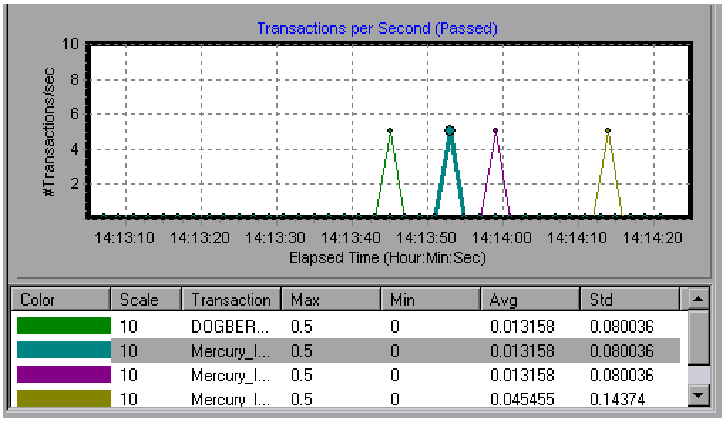
To hide or show a transaction:

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.

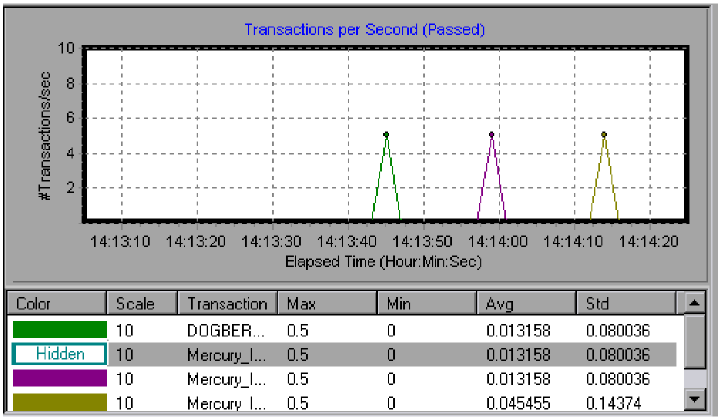
Note that you can also show and hide transactions without opening the Measurement Configuration dialog box, by right-clicking a transaction in the legend and selecting **Show/Hide**.

In the following example, a line is shown for each transaction.



Color	Scale	Transaction	Max	Min	Avg	Std
🟩	10	DOGBER...	0.5	0	0.013158	0.080036
🟦	10	Mercury_I...	0.5	0	0.013158	0.080036
🟪	10	Mercury_I...	0.5	0	0.013158	0.080036
🟨	10	Mercury_I...	0.5	0	0.045455	0.14374

In this example, the second item in the legend is hidden.



Color	Scale	Transaction	Max	Min	Avg	Std
🟩	10	DOGBER...	0.5	0	0.013158	0.080036
Hidden	10	Mercury_I...	0.5	0	0.013158	0.080036
🟪	10	Mercury_I...	0.5	0	0.013158	0.080036
🟨	10	Mercury_I...	0.5	0	0.045455	0.14374

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 Mercury Tuning Console 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.

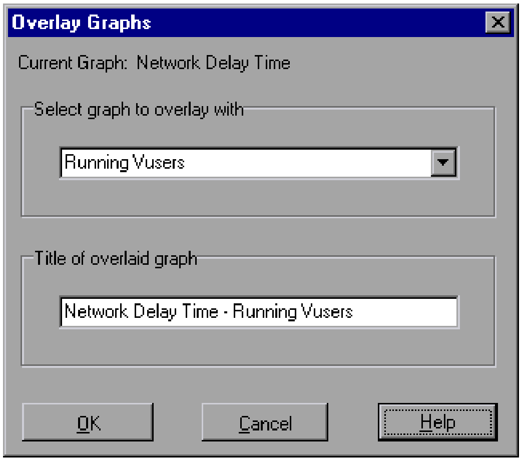
Merging Graphs

Mercury Tuning Console 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.



Overlay Graphs

Current Graph: Network Delay Time

Select graph to overlay with: Running Vusers

Title of overlaid graph: Network Delay Time - Running Vusers

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

Mercury Tuning Console 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

Mercury Tuning Console 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 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 Analysis at the conclusion of the session step, refer to the *LoadRunner Analysis User's Guide*.

Part VII

Remote Tuning

29

Introducing Remote Tuning

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:

- Tuning Agent Specifications
- Supported Operating Systems and Applications
- Remote Tuning Flow

About Tuning Your System from the Console

Mercury Tuning Console'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.

Tuning Agent Specifications

A tuning agent is a multi-threaded, light-weight passive process that performs a task only when requested to do so. The tuning agent's tasks may include managing configuration and server logs, and diagnostic activity.

The tuning agent has no impact on background processes. You can remotely launch and shut down the agent as needed via the Console machine, as long as you have the necessary operating system and tuning agent permissions.

Tuning Agent Requirements

- Java Runtime Environment 1.3 or later
- Hard disk space: 3 MB
- Memory consumption (RAM): 30 MB
- CPU consumption when the tuning agent is not processing a request : 0%

Tuning Agent Security

A tuning agent can be activated and deactivated remotely via the Console, by an authorized administrator.

The tuning agent security is based on the following features:

- Communication between the host machine and the Console machine is performed using a proprietary protocol (OTP) with XML-based payloads over SSL encryption.
- An agent/service-based access control list determines each user's authorizations. Users need to supply a username and password.
- The default guest account has read-only privileges.

Functionality

The Tuning Console provides the following configuration management and diagnostic capabilities:

Management capabilities: These include querying, updating and tracking configuration, and starting and stopping services.

Diagnostic capabilities: These include a network health check, cross-service and host configuration comparisons, scanning configuration settings that do not match recommended values.

Tuning Console Functions

- Validation of multiple system and application configurations (e.g., sanity checks) via a central configuration management console.
- Verification of system and application configuration during a diagnostic session, as a means of determining root cause. (For example, long end-user response times that result in growing request queue sizes can be linked to the threading configuration on an application server).
- Administration (e.g., update configuration, start and stop services) of a wide range of system and application configurations via a central configuration console.

Tuning Agent Functions

- Manages system and application configuration (e.g., retrieves current configuration, updates configuration, starts and stops services, based on client permissions).
- Manages server and application logs (e.g., retrieves, scans, and removes logs, based on client permissions).

Understanding the Configuration Tab

Mercury Tuning Console'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 410.
- Additional buttons that help you tune your system.

Supported Operating Systems and Applications

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

Note: For additional information on requirements and resource usage for tuning agents, see “Host Requirements” on page 389.

Applications

Mercury Tuning Console 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

Remote Tuning Flow

You tune each remote 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.

30

Setting Up Remote Tuning

Before you can tune host machines remotely, you need to perform certain procedures that allow you to connect with them.

This chapter includes the following topics:

- Host Requirements
- Enabling Remote Tuning Through a Firewall
- Configuring Host Connection Parameters
- Connecting to the Host Computer
- Installing a Tuning Agent
- Configuring Special Tuning Agent Settings
- Upgrading a Tuning Agent
- Uninstalling a Tuning Agent

Host Requirements

Before you install and run a tuning agent on the host machine, note the following requirements:

Hard Disk Space

3MB.

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.

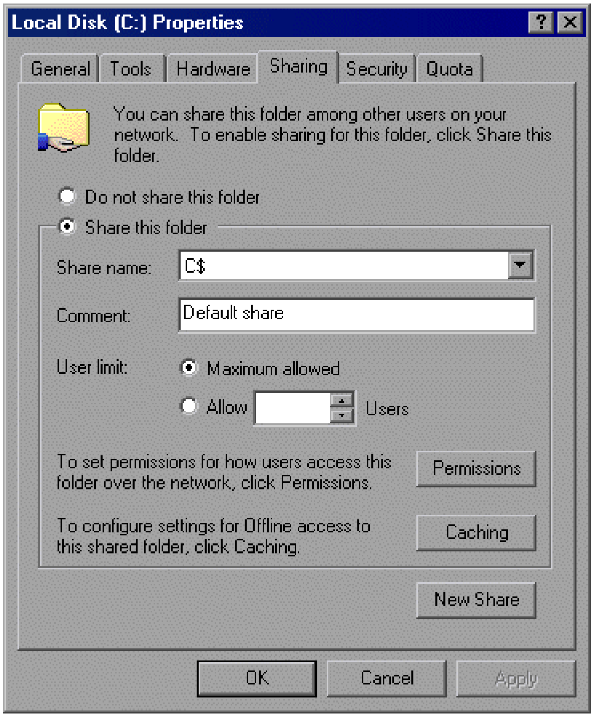
To install WMI, download it from
http://msdn.microsoft.com/library/default.asp?url=/downloads/list/wmi.asp

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:



Local Disk (C:) Properties

You can share this folder among other users on your network. To enable sharing for this folder, click Share this folder.

○ Do not share this folder

● Share this folder

Share name: C$

Comment: Default share

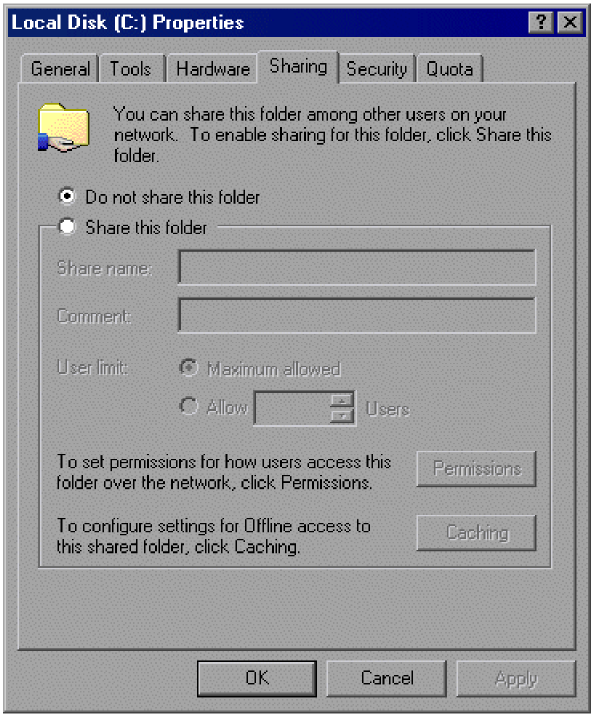
User limit: ● Maximum allowed

○ Allow Users

To set permissions for how users access this folder over the network, click Permissions.

To configure settings for Offline access to this shared folder, click Caching.

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.



You can share this folder among other users on your network. To enable sharing for this folder, click Share this folder.

● Do not share this folder

○ Share this folder

Share name:

Comment:

User limit: ● Maximum allowed

○ Allow Users

To set permissions for how users access this folder over the network, click Permissions.

To configure settings for Offline access to this shared folder, click Caching.

If this is the case, click **Share this folder** and enter C$ in the **Share name** list.

3 Click **OK**.

Enabling Remote 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 Tuning Agent Settings
3. Specifying Operating System Settings

Note: In addition, you can optionally specify SNMP settings.

---

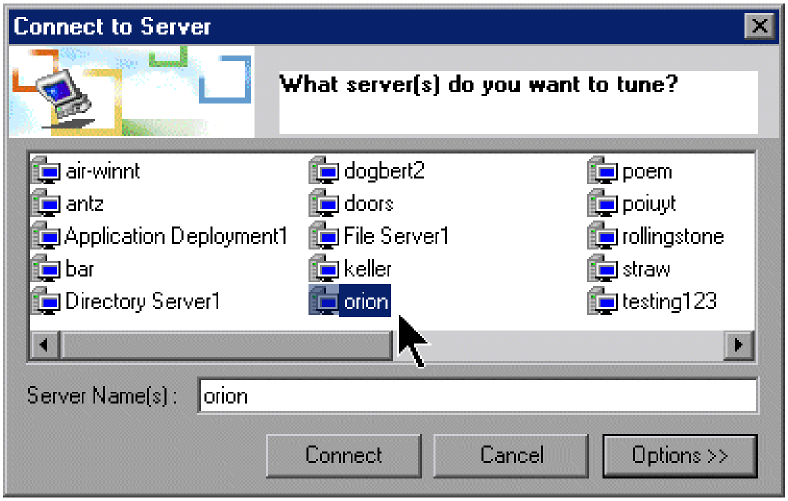
4. Connecting to the Host Computer

Choosing the Host Machine

To choose the host that you want to tune (if the host does not appear in the Servers Configuration tree):

1. 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.



Connect to Server

What server(s) do you want to tune?

Server Name(s) : orion

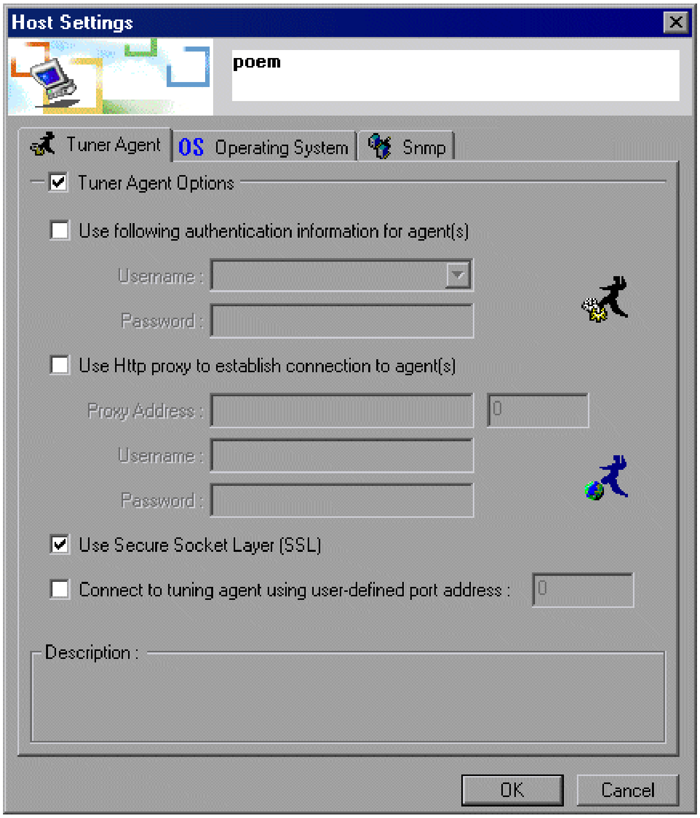
2 Choose the host that you want to tune.

Specifying Tuning Agent Settings

Before you can connect to the host machine, you need to specify the tuning 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 tuning 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.



Host Settings

poem

☑ Tuner Agent Options

☐ Use following authentication information for agent(s)

Username :

Password :

☐ Use Http proxy to establish connection to agent(s)

Proxy Address : 0

Username :

Password :

☑ Use Secure Socket Layer (SSL)

☐ Connect to tuning agent using user-defined port address : 0

Description :

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. Mercury Tuning Console 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, Mercury Tuning Console uses the *guest* username, allowing you only to query the host.

Note: For details of how to change passwords, see “Changing Tuning Agent Passwords” on page 434.

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.

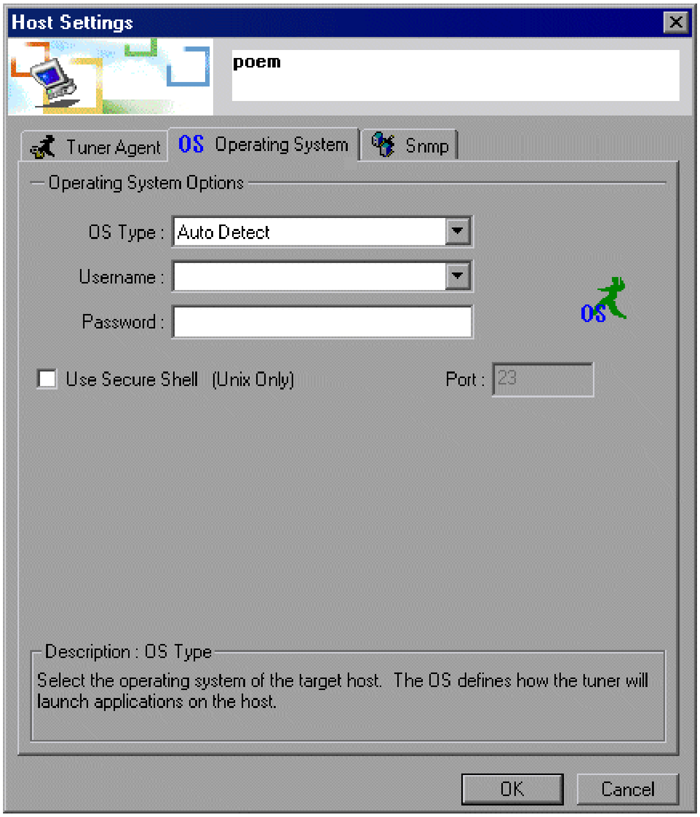
5 Check the **Use Secure Socket Layer (SSL)** box to use SSL for all connections (recommended).

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 tuning 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.



poem

OS Type : Auto Detect

Username :

Password :

□ Use Secure Shell (Unix Only) Port : 23

Description : OS Type

Select the operating system of the target host. The OS defines how the tuner will launch applications on the host.

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.

   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. For information on specifying SNMP settings, see "Configuring Special Tuning Agent Settings" on page 402.

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.

Installing a Tuning Agent

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), you can install and start the tuning agent by using one of the following methods:

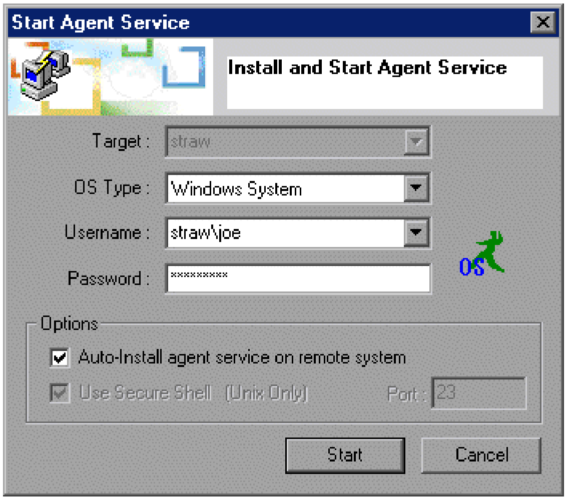
- Remote installation of the tuning agent
- Local installation of the tuning agent on the host machine from the Mercury LoadRunner CD

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:



Start Agent Service

Install and Start Agent Service

Target : straw

OS Type : Windows System

Username : straw\joe

Password : *********

Options

☑ Auto-Install agent service on remote system

☑ Use Secure Shell (Unix Only) Port : 23

2 Specify the host's OS Type. To let Mercury Tuning Console 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 have configured the settings correctly, Mercury Tuning Console installs the tuning agent on the host and starts the pe_agent.exe process on it.

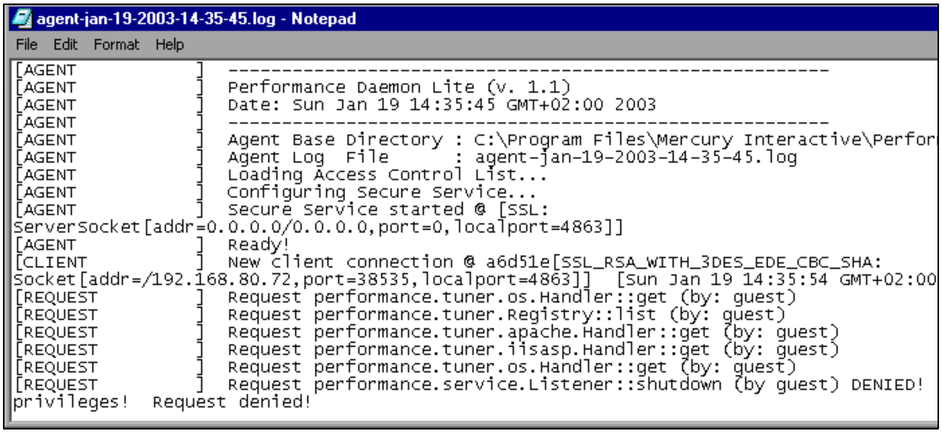
Note: On a Windows host, the tuning agent is installed under C:\Program Files\Mercury Interactive\Tuning Agent. On a UNIX host, it is installed under /tmp/tuningagent.

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:



```
[AGENT          ]  ---------------------------------------------------------
[AGENT          ]  Performance Daemon Lite (v. 1.1)
[AGENT          ]  Date: Sun Jan 19 14:35:45 GMT+02:00 2003
[AGENT          ]  ---------------------------------------------------------
[AGENT          ]  Agent Base Directory : C:\Program Files\Mercury Interactive\Perfor
[AGENT          ]  Agent Log  File      : agent-jan-19-2003-14-35-45.log
[AGENT          ]  Loading Access Control List...
[AGENT          ]  Configuring Secure Service...
[AGENT          ]  Secure Service started @ [SSL:
ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=4863]]
[AGENT          ]  Ready!
[CLIENT         ]  New client connection @ a6d51e[SSL_RSA_WITH_3DES_EDE_CBC_SHA:
Socket[addr=/192.168.80.72,port=38535,localport=4863]]  [Sun Jan 19 14:35:54 GMT+02:00
[REQUEST        ]  Request performance.tuner.os.Handler::get (by: guest)
[REQUEST        ]  Request performance.tuner.Registry::list (by: guest)
[REQUEST        ]  Request performance.tuner.apache.Handler::get (by: guest)
[REQUEST        ]  Request performance.tuner.iisasp.Handler::get (by: guest)
[REQUEST        ]  Request performance.tuner.os.Handler::get (by: guest)
[REQUEST        ]  Request performance.service.Listener::shutdown (by guest) DENIED!
privileges!  Request denied!
```

If the tuning agent does not start, or does not display the expected information, you may need to configure its settings. See "Configuring Tuning Agents" on page 433.

Local Installation of a Tuning Agent

You can install a tuning agent locally on a host machine using one of the following methods:

- From the Mercury LoadRunner CD (for Windows hosts only)
- By copying a file from the Console machine to the host machine and executing it

To install a tuning agent locally on a Windows host from the Mercury LoadRunner Add-ins CD:

➤ Choose the **Tuning 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** > **Tuning Agent** on the host machine.

To install a tuning agent locally on a Windows host by copying a file from the Console machine:

1 Extract the perfagent.tar file (located in the *\Mercury LoadRunner\console\bin* directory on the Console machine) to a directory on the host machine (for example, to
C:\Program Files\Mercury Interactive\Tuning Agent).

2 On the host machine, set the PE_HOME environment variable (in this example,
PE_HOME = C:\Program Files\Mercury Interactive\Tuning Agent).

3 On the host machine, launch the tuning agent application. In this example, you would run the following file:

```
C:\Program Files\Mercury Interactive\Tuning Agent\agent\bin\pe_agent.exe
```

To install a tuning agent locally on a UNIX host:

1 Copy the perfagent.tar file (from the *\Mercury LoadRunner\console\bin* directory on the Windows machine where the Console is installed) to a directory on the UNIX machine (for example, to */usr/local/tuningagent*), and extract it to that directory.

2 On the host machine, set the PE_HOME environment variable (in this example for csh,
setenv PE_HOME /usr/local/tuningagent)

3 On the host machine, launch the tuning agent batch file. In this example, you would run the following file:

```
/usr/local/tuningagent/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/tuningagent
```

Configuring Special Tuning Agent Settings

You can configure the following special types of tuning agent settings:

➤ SNMP Settings

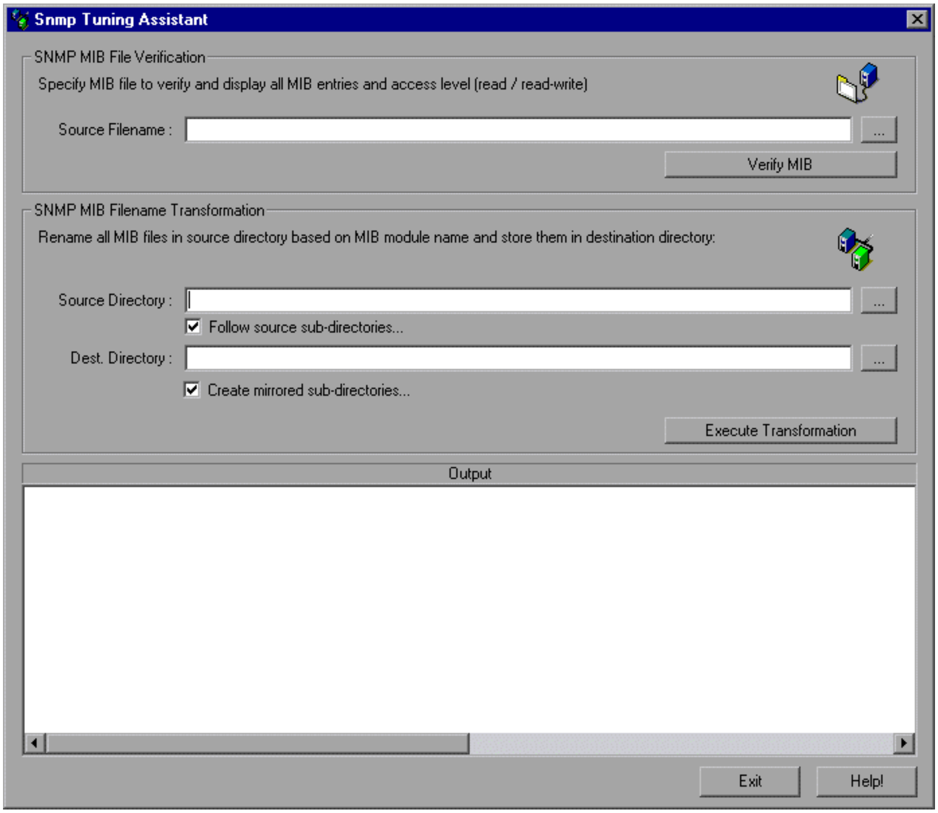
Enabling SNMP For Tuning

Mercury Tuning Console 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 Select **Start** > **Programs** > **Mercury LoadRunner** > **Tools** > **SNMP Tuning Assistant**.

The SNMP Tuning Assistant dialog box is displayed.



Snmp Tuning Assistant

SNMP MIB File Verification

Specify MIB file to verify and display all MIB entries and access level (read / read-write)

Source Filename :

SNMP MIB Filename Transformation

Rename all MIB files in source directory based on MIB module name and store them in destination directory:

Source Directory :

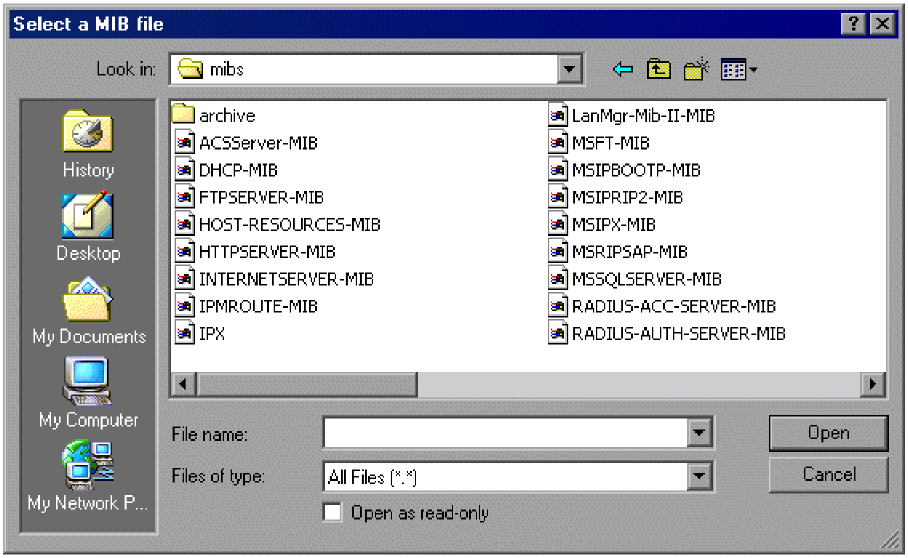
☑ Follow source sub-directories...

Dest. Directory :

☑ Create mirrored sub-directories...

Output

2 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.



Select a MIB file

Look in: mibs

archive
ACSServer-MIB
DHCP-MIB
FTPSERVER-MIB
HOST-RESOURCES-MIB
HTTPSERVER-MIB
INTERNETSERVER-MIB
IPMROUTE-MIB
IPX
LanMgr-Mib-II-MIB
MSFT-MIB
MSIPBOOTP-MIB
MSIPRIP2-MIB
MSIPX-MIB
MSRIPSAP-MIB
MSSQLSERVER-MIB
RADIUS-ACC-SERVER-MIB
RADIUS-AUTH-SERVER-MIB

File name:

Files of type: All Files (*.*)

□ Open as read-only

3 Select the file for the protocol that you want to use, and click **Open**. The filename now appears in the Source Filename field.

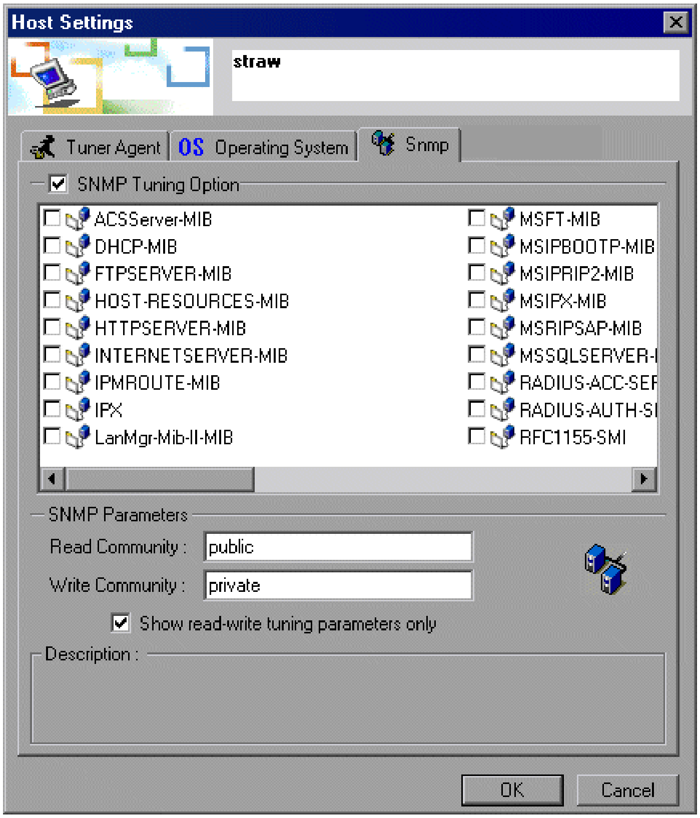
4 Click **Verify** to check that the MIB file has not been corrupted.

5 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.

6 When the file has been successfully verified, click **Exit** to close the SNMP Tuning 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**.



straw

☑ SNMP Tuning Option

□ ACSServer-MIB
□ DHCP-MIB
□ FTPSERVER-MIB
□ HOST-RESOURCES-MIB
□ HTTPSERVER-MIB
□ INTERNETSERVER-MIB
□ IPMROUTE-MIB
□ IPX
□ LanMgr-Mib-II-MIB
□ MSFT-MIB
□ MSIPBOOTP-MIB
□ MSIPRIP2-MIB
□ MSIPX-MIB
□ MSRIPSAP-MIB
□ MSSQLSERVER-I
□ RADIUS-ACC-SEF
□ RADIUS-AUTH-SI
□ RFC1155-SMI

SNMP Parameters

Read Community : public

Write Community : private

☑ Show read-write tuning parameters only

Description :

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.

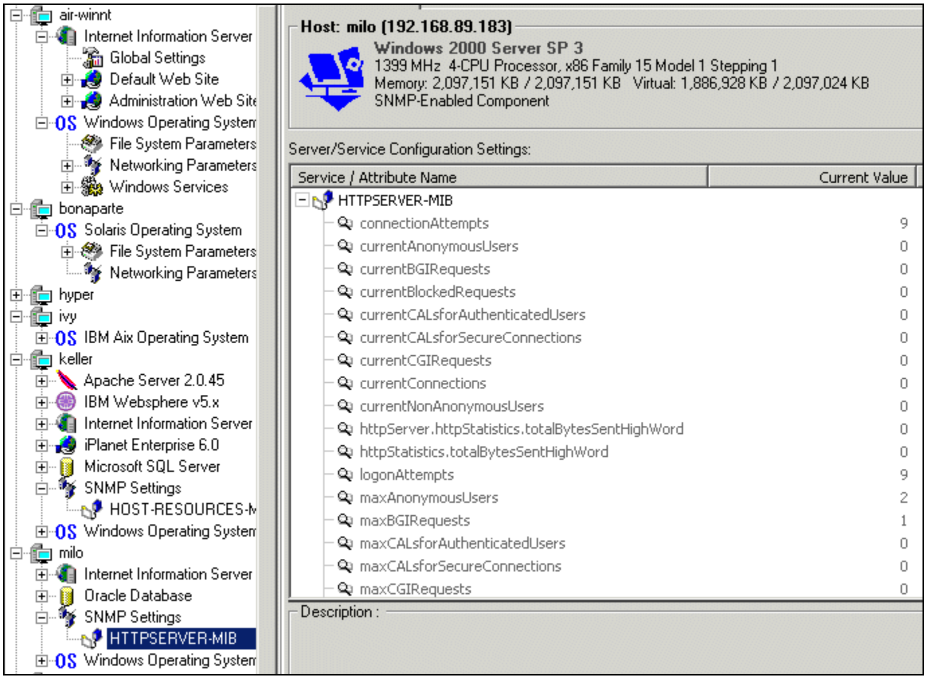
5 In the Write Community field, specify the community to use for writing new values to the server. Default: private.

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.



- air-winnt
  - Internet Information Server
    - Global Settings
    - Default Web Site
    - Administration Web Site
  - OS Windows Operating Systen
    - File System Parameters
    - Networking Parameters
    - Windows Services
- bonaparte
  - OS Solaris Operating System
    - File System Parameters
    - Networking Parameters
- hyper
- ivy
  - OS IBM Aix Operating System
- keller
  - Apache Server 2.0.45
  - IBM Websphere v5.x
  - Internet Information Server
  - iPlanet Enterprise 6.0
  - Microsoft SQL Server
  - SNMP Settings
    - HOST-RESOURCES-M
  - OS Windows Operating Systen
- milo
  - Internet Information Server
  - Oracle Database
  - SNMP Settings
    - HTTPSERVER-MIB
  - OS Windows Operating Systen

Host: milo (192.168.89.183)

Windows 2000 Server SP 3
1399 MHz 4-CPU Processor, x86 Family 15 Model 1 Stepping 1
Memory: 2,097,151 KB / 2,097,151 KB Virtual: 1,886,928 KB / 2,097,024 KB
SNMP-Enabled Component

Server/Service Configuration Settings:

Service / Attribute Name	Current Value
HTTPSERVER-MIB	
connectionAttempts	9
currentAnonymousUsers	0
currentBGIRequests	0
currentBlockedRequests	0
currentCALsforAuthenticatedUsers	0
currentCALsforSecureConnections	0
currentCGIRequests	0
currentConnections	0
currentNonAnonymousUsers	0
httpServer.httpStatistics.totalBytesSentHighWord	0
httpStatistics.totalBytesSentHighWord	0
logonAttempts	9
maxAnonymousUsers	2
maxBGIRequests	1
maxCALsforAuthenticatedUsers	0
maxCALsforSecureConnections	0
maxCGIRequests	0

Description :

Upgrading a Tuning Agent

To upgrade a tuning agent with a new version:

1. Stop the tuning agent (see “Stopping a Tuning Agent” on page 452).
2. Remove the tuning agent from the host machine (see the next section).
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 “Installing a Tuning Agent” on page 398.

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** > **Tuning Agent** > **Uninstall Tuning Agent 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 “Stopping a Tuning Agent” on page 452).
2. On the host machine, delete the tuning agent files from the appropriate location:

➤ On a Windows machine, delete the directory C:\Mercury Interactive\Tuning Agent\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/tuningagent.

31

Viewing and Changing Host Information

Once you have set up your connection with the host machines, the tuning agents let you view information about the machines and their services. They also allow you to change properties on the host machines where applicable.

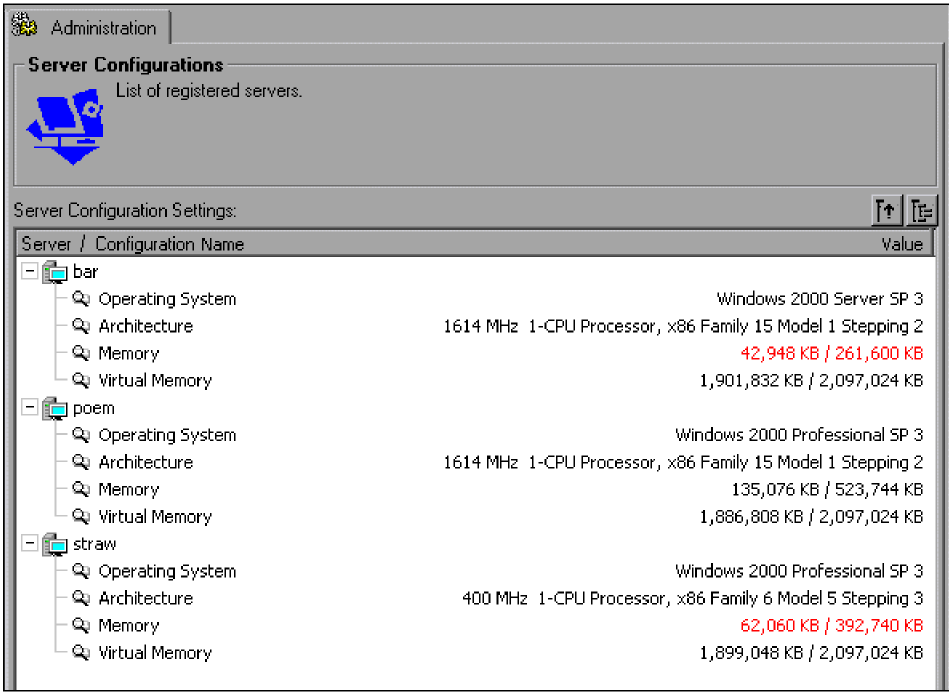
This chapter includes the following topics:

- Viewing the Host Information
- Using Expert Mode
- Viewing Windows Services
- Viewing Differences Only
- Comparing Hosts and Services
- Viewing and Changing Tuning Parameter Values
- Updating the Host or Service with Changes
- Changing Recommended Values

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 have 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:



Server Configurations

List of registered servers.

Server Configuration Settings:

Server / Configuration Name	Value
bar	
Operating System	Windows 2000 Server SP 3
Architecture	1614 MHz 1-CPU Processor, x86 Family 15 Model 1 Stepping 2
Memory	42,948 KB / 261,600 KB
Virtual Memory	1,901,832 KB / 2,097,024 KB
poem	
Operating System	Windows 2000 Professional SP 3
Architecture	1614 MHz 1-CPU Processor, x86 Family 15 Model 1 Stepping 2
Memory	135,076 KB / 523,744 KB
Virtual Memory	1,886,808 KB / 2,097,024 KB
straw	
Operating System	Windows 2000 Professional SP 3
Architecture	400 MHz 1-CPU Processor, x86 Family 6 Model 5 Stepping 3
Memory	62,060 KB / 392,740 KB
Virtual Memory	1,899,048 KB / 2,097,024 KB

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 100 MB 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**.

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:

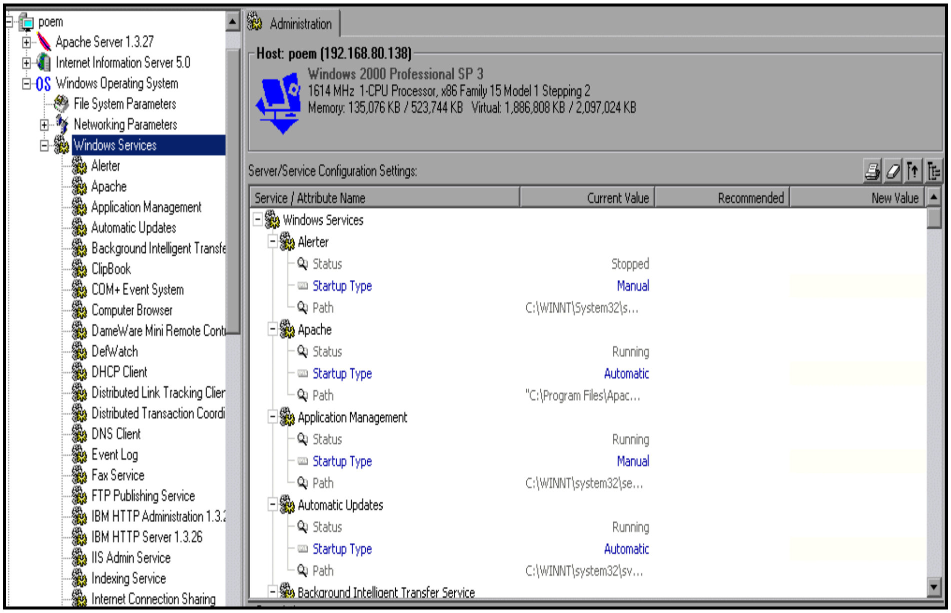
Choose **Normal View** from the box in the tool bar, or right-click the host or service and choose **View Mode** > **Normal View**. This disables Expert mode for the selected host or service.

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 449.

Expanding the Windows Services element displays the host machine's services.



- poem
  - Apache Server 1.3.27
  - Internet Information Server 5.0
  - OS Windows Operating System
    - File System Parameters
    - Networking Parameters
    - Windows Services
      - Alerter
      - Apache
      - Application Management
      - Automatic Updates
      - Background Intelligent Transfe
      - ClipBook
      - COM+ Event System
      - Computer Browser
      - DameWare Mini Remote Contr
      - DefWatch
      - DHCP Client
      - Distributed Link Tracking Clier
      - Distributed Transaction Coordi
      - DNS Client
      - Event Log
      - Fax Service
      - FTP Publishing Service
      - IBM HTTP Administration 1.3.2
      - IBM HTTP Server 1.3.26
      - IIS Admin Service
      - Indexing Service
      - Internet Connection Sharing

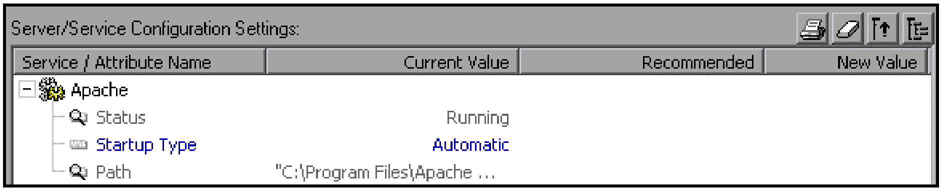
Host: poem (192.168.80.138)

Windows 2000 Professional SP 3
1614 MHz 1-CPU Processor, x86 Family 15 Model 1 Stepping 2
Memory: 135,076 KB / 523,744 KB Virtual: 1,886,808 KB / 2,097,024 KB

Server/Service Configuration Settings:

Service / Attribute Name	Current Value	Recommended	New Value
Windows Services			
Alerter			
Status	Stopped		
Startup Type	Manual		
Path	C:\WINNT\System32\s...		
Apache			
Status	Running		
Startup Type	Automatic		
Path	"C:\Program Files\Apac...		
Application Management			
Status	Running		
Startup Type	Manual		
Path	C:\WINNT\system32\se...		
Automatic Updates			
Status	Running		
Startup Type	Automatic		
Path	C:\WINNT\system32\sv...		
Background Intelligent Transfer Service			

When you click a service in the Windows Services tree, Mercury Tuning Console displays the service's status in the Configuration tab.



Server/Service Configuration Settings:

Service / Attribute Name	Current Value	Recommended	New Value
Apache			
Status	Running		
Startup Type	Automatic		
Path	"C:\Program Files\Apache ...		

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 416).

- Path—The path to the 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

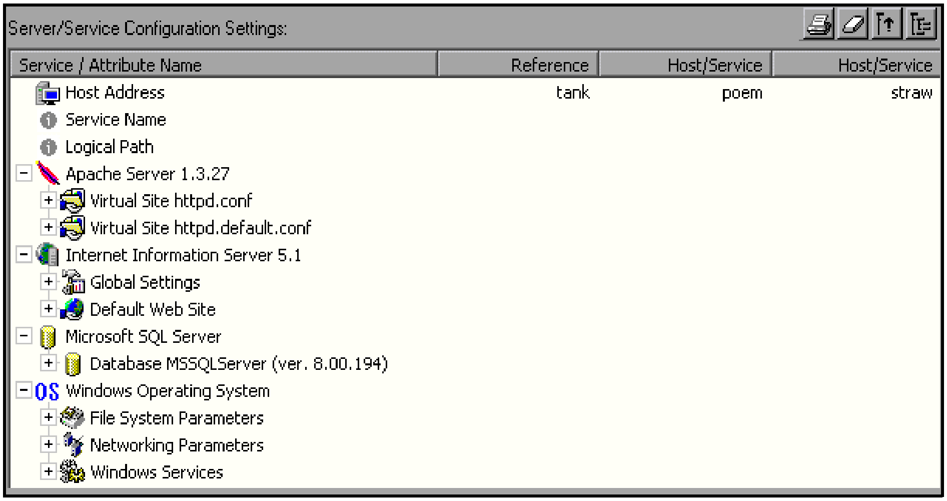
Mercury Tuning Console allows you to compare the values of properties of different hosts and services, and shows you the properties that are different.

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.

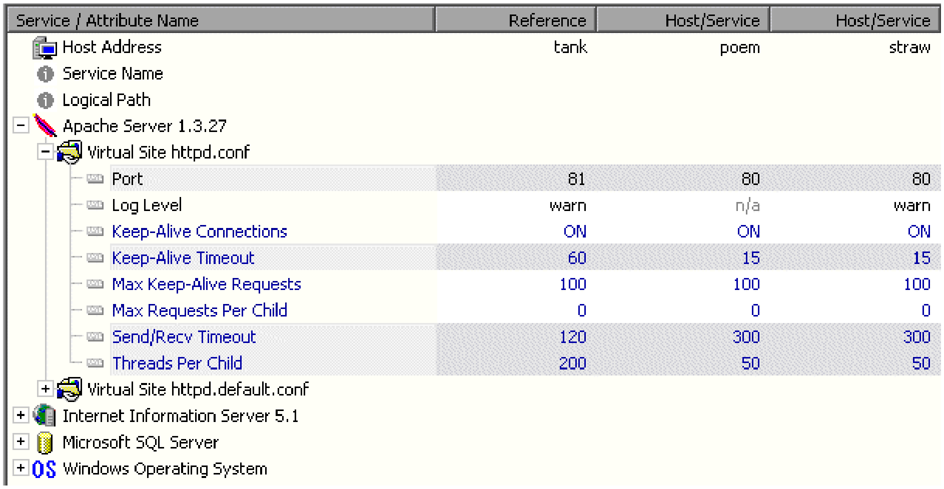
3 Click **Compare Configuration Settings**. The Configuration tab shows a column of properties for each of the servers or services being compared.



Server/Service Configuration Settings:

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			
+ Virtual Site httpd.default.conf			
− Internet Information Server 5.1			
+ Global Settings			
+ Default Web Site			
− Microsoft SQL Server			
+ Database MSSQLServer (ver. 8.00.194)			
− OS Windows Operating System			
+ File System Parameters			
+ Networking Parameters			
+ Windows Services			

4 Expand the service whose properties you want to view, or click **Expand All Nodes** to expand all the services. Mercury Tuning Console 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, Mercury Tuning Console displays the following:

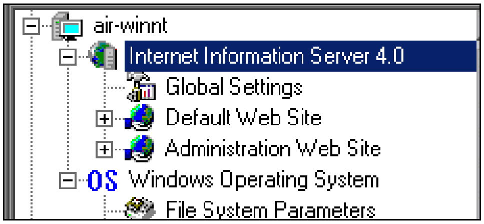
- **Current value**
- **Recommended value**—the value recommended by Mercury Tuning Console
- **New value**—the value that you entered (if you changed the old value)

The different parameters are color-coded as follows:

Color	Meaning
blue	important
black	for advanced users
red	critical tuning parameter
gray	read-only

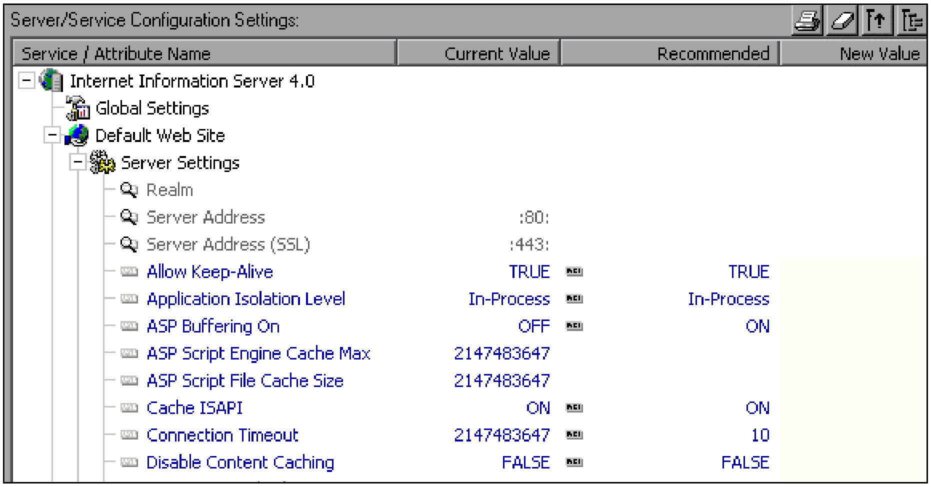
To change the value of a host parameter:

1 Select the host or service that you want to tune, by clicking the relevant icon in the Server Configuration tree.



air-winnt
Internet Information Server 4.0
Global Settings
Default Web Site
Administration Web Site
OS Windows Operating System
File System Parameters

In the Administration tab, Mercury Tuning Console displays the parameters relevant to the selected host or service.

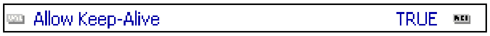


Server/Service Configuration Settings:

Service / Attribute Name	Current Value	Recommended	New Value
Internet Information Server 4.0			
Global Settings			
Default Web Site			
Server Settings			
Realm			
Server Address	:80:		
Server Address (SSL)	:443:		
Allow Keep-Alive	TRUE	TRUE	
Application Isolation Level	In-Process	In-Process	
ASP Buffering On	OFF	ON	
ASP Script Engine Cache Max	2147483647		
ASP Script File Cache Size	2147483647		
Cache ISAPI	ON	ON	
Connection Timeout	2147483647	10	
Disable Content Caching	FALSE	FALSE	

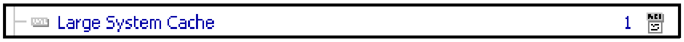
The values in the Recommended column are based on information in Mercury Tuning Console's knowledge base.

If the Tuning Console'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:



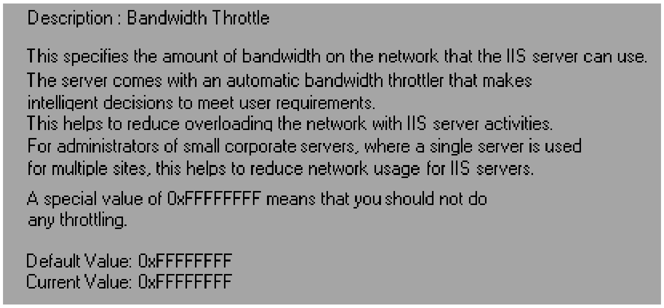
Allow Keep-Alive TRUE

If the Tuning Console'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:



Large System Cache 1

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.



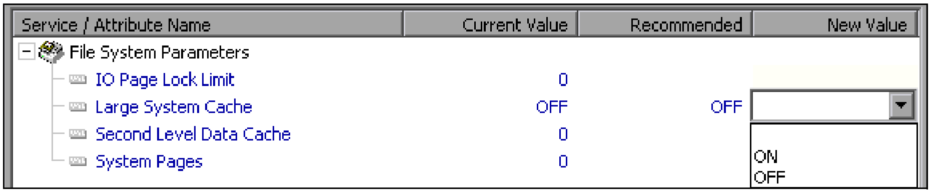
Description : Bandwidth Throttle

This specifies the amount of bandwidth on the network that the IIS server can use.
The server comes with an automatic bandwidth throttler that makes
intelligent decisions to meet user requirements.
This helps to reduce overloading the network with IIS server activities.
For administrators of small corporate servers, where a single server is used
for multiple sites, this helps to reduce network usage for IIS servers.

A special value of 0xFFFFFFFF means that you should not do
any throttling.

Default Value: 0xFFFFFFFF
Current Value: 0xFFFFFFFF

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.



Service / Attribute Name	Current Value	Recommended	New Value
⊟ File System Parameters			
IO Page Lock Limit	0		
Large System Cache	OFF	OFF	▾
Second Level Data Cache	0		
System Pages	0		

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" in the next section). 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

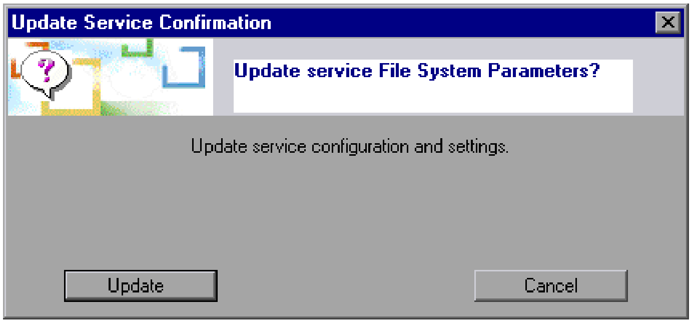
Changes that you make to parameter values do not take effect until you update the relevant host or service.

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.



Update Service Confirmation

Update service File System Parameters?

Update service configuration and settings.

2 Click **Update** to update the selected host or service. Note that some services need to be restarted for the changes to take effect.

Mercury Tuning Console 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

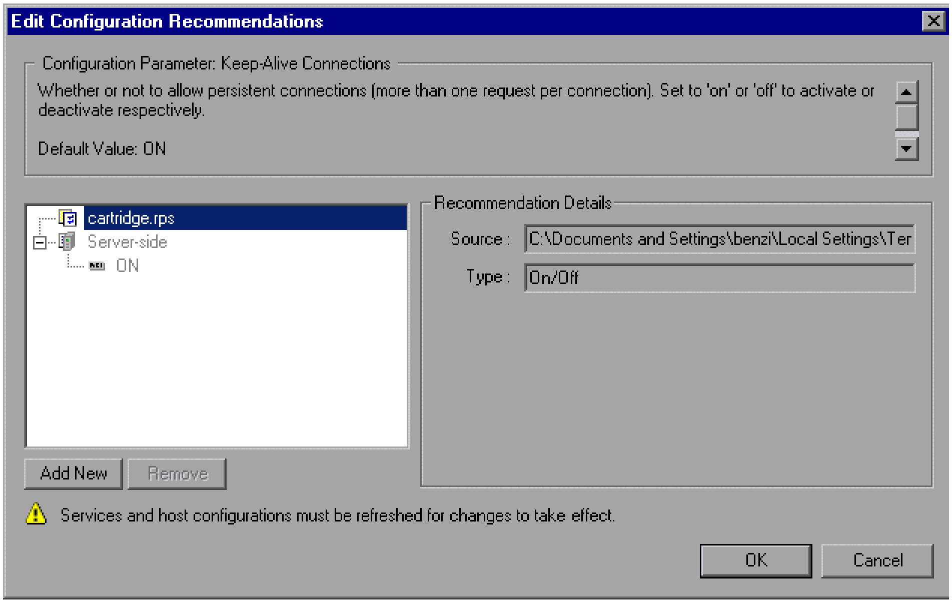
Mercury Tuning Console allows you to add a value to a parameter's list of recommended values, or to remove a value from the list. You can also edit an existing value.

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:

1 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 Recommendations window is displayed.



Edit Configuration Recommendations

Configuration Parameter: Keep-Alive Connections

Whether or not to allow persistent connections (more than one request per connection). Set to 'on' or 'off' to activate or deactivate respectively.

Default Value: ON

Recommendation Details

Source : C:\Documents and Settings\benzi\Local Settings\Ter

Type : On/Off

Services and host configurations must be refreshed for changes to take effect.

2 Use the **Add New** and **Remove** buttons to add or remove recommended values, respectively.

Understanding the Edit Configuration Recommendations Window

You use the Edit Configuration Recommendations window to modify a parameter's list of recommended values.

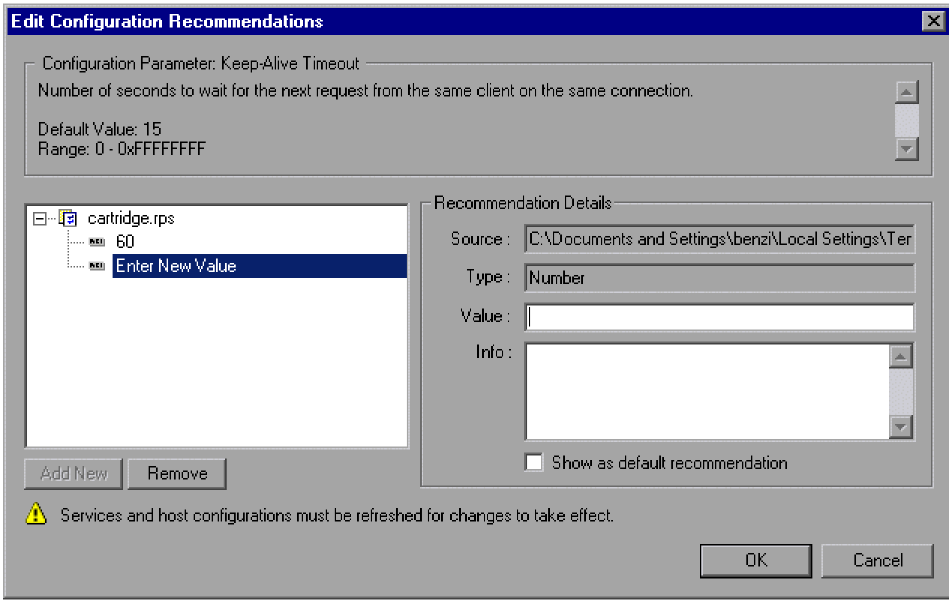
If there are recommended values for a selected parameter, they are displayed in the left pane.

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.



Configuration Parameter: Keep-Alive Timeout

Number of seconds to wait for the next request from the same client on the same connection.

Default Value: 15
Range: 0 - 0xFFFFFFFF

cartridge.rps
60
Enter New Value

Source : C:\Documents and Settings\benzi\Local Settings\Ter

Type : Number

Value :

Info :

□ Show as default recommendation

Services and host configurations must be refreshed for changes to take effect.

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 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
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.

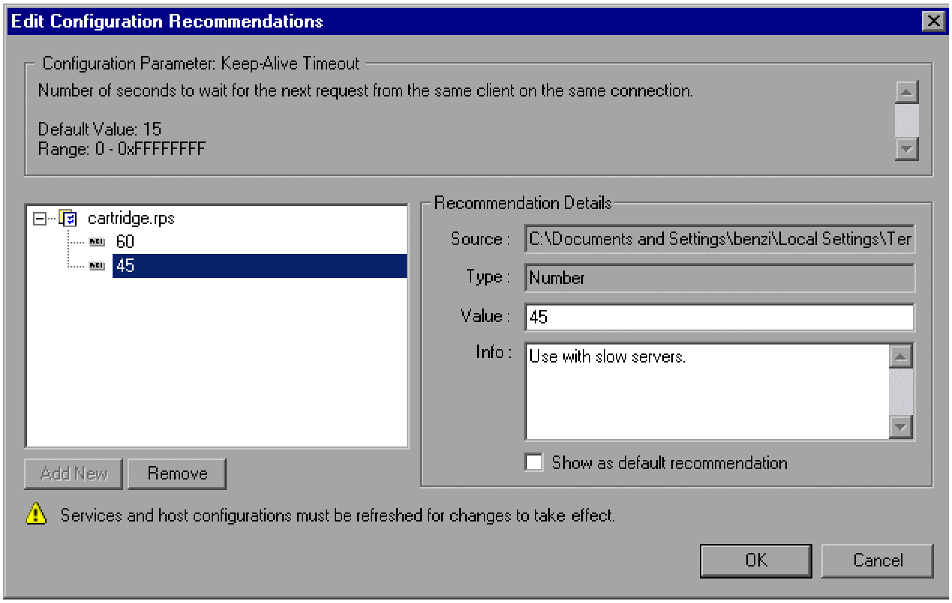
4 To cause the new value to be displayed as the recommended value (that is, as the first value in the list), check **Show as default recommendation**.

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 Recommendations window, select the value that you want to remove, as in the following example:



Edit Configuration Recommendations

Configuration Parameter: Keep-Alive Timeout

Number of seconds to wait for the next request from the same client on the same connection.

Default Value: 15
Range: 0 - 0xFFFFFFFF

cartridge.rps
60
45

Recommendation Details

Source : C:\Documents and Settings\benzi\Local Settings\Ter

Type : Number

Value : 45

Info : Use with slow servers.

□ Show as default recommendation

Services and host configurations must be refreshed for changes to take effect.

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.

32

Exporting and Importing Configuration Settings

This chapter describes how to export and import configuration settings.

It includes the following topics:

- Exporting Configuration Settings
- Importing Configuration Settings
- 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 Configuration Settings

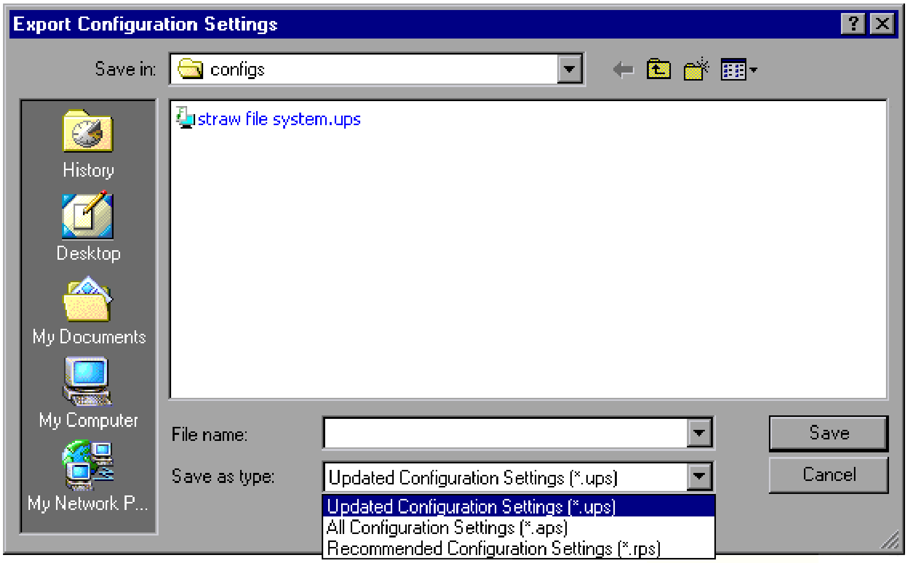
The Export function allows you to store the configuration settings of the entire host, or only a subset of these settings (for example, the settings for a specific service). You can export the following sets of values:

- 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, Mercury Tuning Console handles them as follows:
  - When you import a recommended settings file (a file with the *.rps* extension), Mercury Tuning Console inserts the values in the Recommended column.
  - When you import an updated settings file , (a file with the *.ups* extension), Mercury Tuning Console inserts the values in the New Value column.

See "Updating the Host or Service with Changes" on page 419 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 Configuration Settings dialog box is displayed.



Export Configuration Settings

Save in: configs

straw file system.ups

File name:

Save as type: Updated Configuration Settings (*.ups)

Updated Configuration Settings (*.ups)
All Configuration Settings (*.aps)
Recommended Configuration Settings (*.rps)

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, Mercury Tuning Console 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**.

Mercury Tuning Console saves your settings in a file with the name you specified.

Importing Configuration Settings

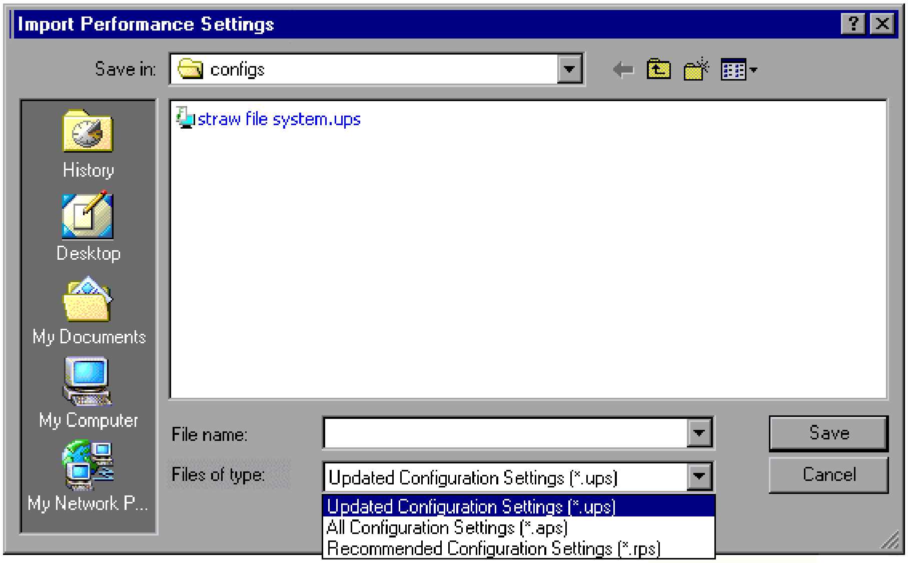
The Import function allows you to import previously stored settings and apply them to hosts and services. You can import and apply settings to:

- 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, Mercury Tuning Console 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.



Save in: configs

straw file system.ups

File name:

Files of type: Updated Configuration Settings (*.ups)

Import Performance Settings
?
History
Desktop
My Documents
My Computer
My Network P...
Save
Cancel
Updated Configuration Settings (*.ups)
All Configuration Settings (*.aps)
Recommended Configuration Settings (*.rps)

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**.

Mercury Tuning Console 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 419). 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

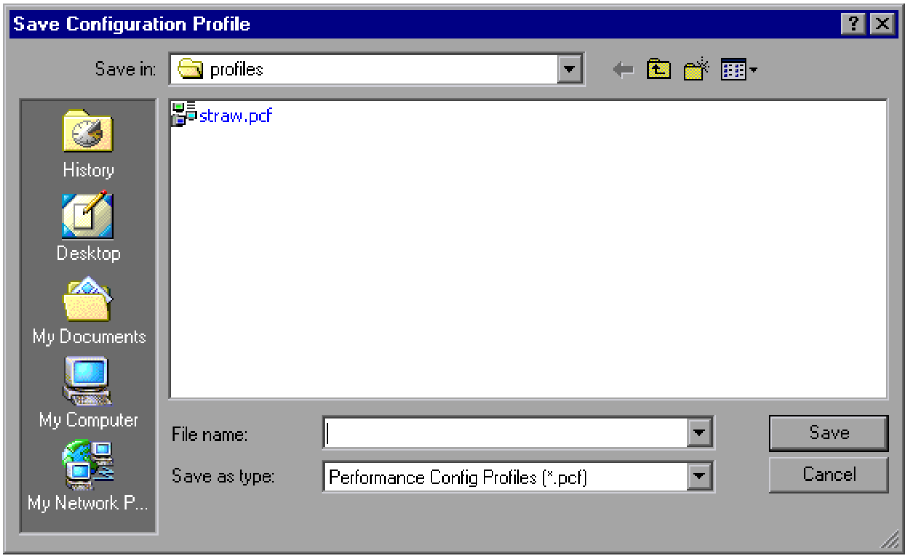
A profile contains a list of hosts.

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, Mercury Tuning Console 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.



Save Configuration Profile

Save in: profiles

straw.pcf

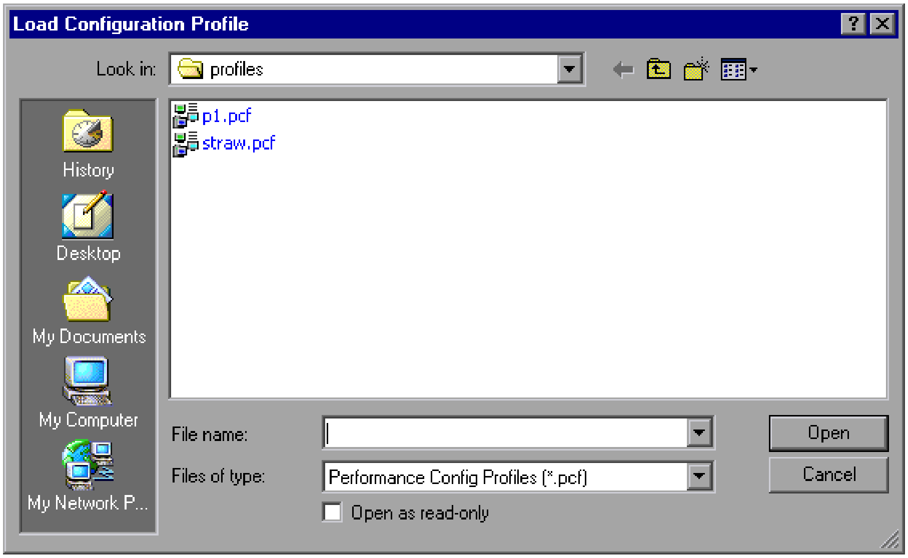
File name:

Save as type: Performance Config Profiles (*.pcf)

2 Enter a name for your profile and click **Save**. Mercury Tuning Console 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.



Load Configuration Profile

Look in: profiles

p1.pcf
straw.pcf

File name:

Files of type: Performance Config Profiles (*.pcf)

□ Open as read-only

2 Choose the profile you want to load, and click **Open**. Mercury Tuning Console 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. Mercury Tuning Console displays the new configuration in the Configuration tab.

Creating a New Profile

Note: When Mercury Tuning Console 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**. Mercury Tuning Console 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.

33

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 Mercury Tuning Console workstation, or locally on the target system (see “Connecting to the Host Computer” on page 398).

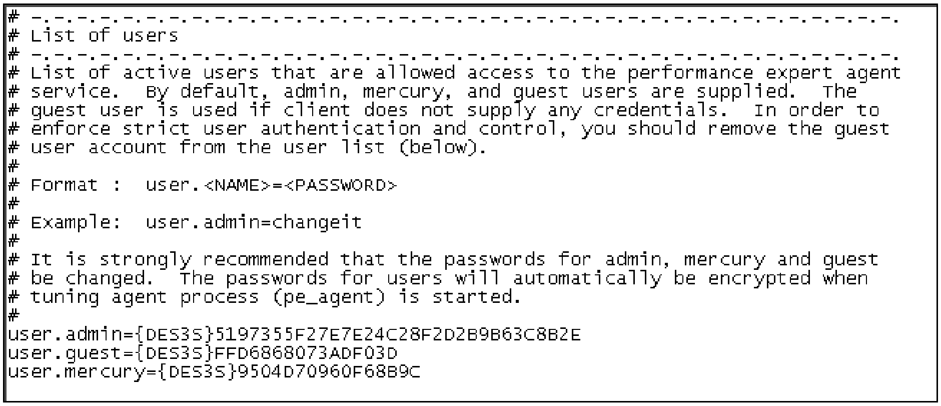
The tuning agent does not require any registry updates. It uses the PE_HOME, PE_USE_SSL, and PE_USE_PORT optional environment variables. Alternatively, you can pass these as arguments to the pe_agent and pe_registry commands.

Changing Tuning Agent Passwords

The tuning agent is supplied with a number of predefined users and passwords (see the table in “Specifying Tuning Agent Settings,” on page 395). 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 Tuning Agent\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 agent
# 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.exe (Windows) and pe_agent (UNIX) files on the host.

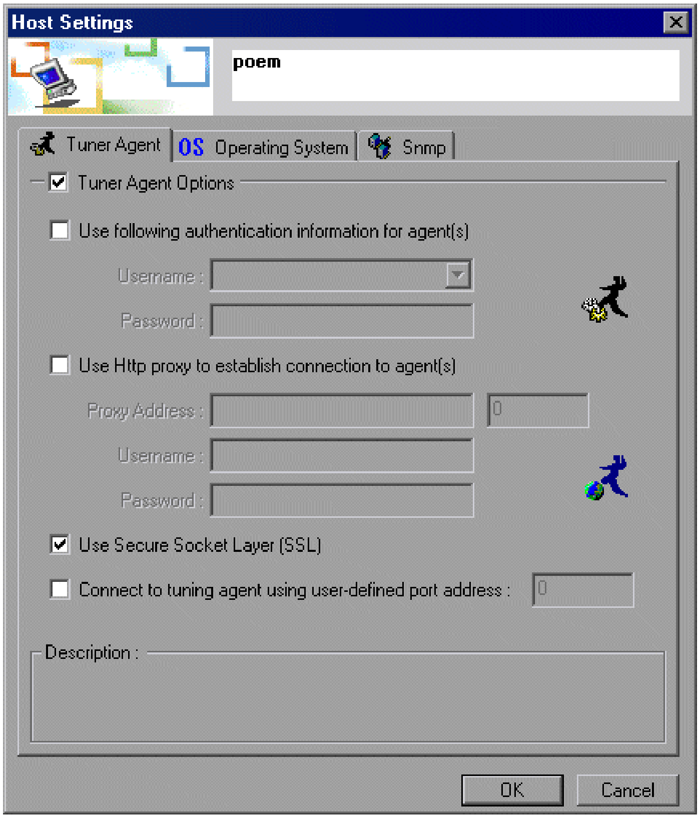
Tip: On a Windows host, the pe_agent.exe file is located in the *Program Files*\Mercury Interactive\Tuning Agent\agent\bin directory.

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:



Host Settings

poem

Tuner Agent | OS Operating System | Snmp

☑ Tuner Agent Options

☐ Use following authentication information for agent(s)

Username :

Password :

☐ Use Http proxy to establish connection to agent(s)

Proxy Address : 0

Username :

Password :

☑ Use Secure Socket Layer (SSL)

☐ Connect to tuning agent using user-defined port address : 0

Description :

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:\Mercury Tuning Console	Launches the tuning agent at port 1236 with SSL enabled, using configuration and tuners from C:\Mercury Tuning Console.

➤ 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:\Mercury Tuning Console
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.

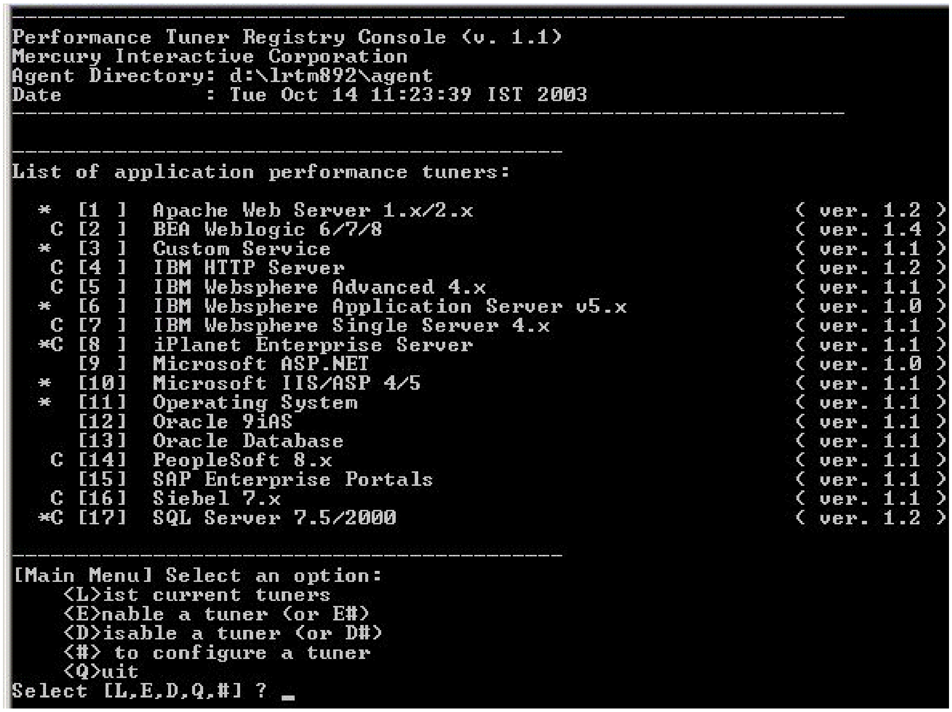
The Performance Tuner Registry provides a command-line interface for configuring the individual tuners.

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, Mercury Tuning Console 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** > **Tuning Agent** > **Configuration**.

Mercury Tuning Console invokes the Performance Tuner Registry and displays the Performance Tuner Registry Console:



```
-------------------------------------------------------------------
Performance Tuner Registry Console (v. 1.1)
Mercury Interactive Corporation
Agent Directory: d:\lrtm892\agent
Date           : Tue Oct 14 11:23:39 IST 2003
-------------------------------------------------------------------

--------------------------------------------
List of application performance tuners:

  *  [1 ]  Apache Web Server 1.x/2.x                      ( ver. 1.2 )
   C [2 ]  BEA Weblogic 6/7/8                             ( ver. 1.4 )
  *  [3 ]  Custom Service                                 ( ver. 1.1 )
   C [4 ]  IBM HTTP Server                                ( ver. 1.2 )
   C [5 ]  IBM Websphere Advanced 4.x                     ( ver. 1.1 )
  *  [6 ]  IBM Websphere Application Server v5.x          ( ver. 1.0 )
   C [7 ]  IBM Websphere Single Server 4.x                ( ver. 1.1 )
  *C [8 ]  iPlanet Enterprise Server                      ( ver. 1.1 )
     [9 ]  Microsoft ASP.NET                              ( ver. 1.0 )
  *  [10]  Microsoft IIS/ASP 4/5                          ( ver. 1.1 )
  *  [11]  Operating System                               ( ver. 1.1 )
     [12]  Oracle 9iAS                                    ( ver. 1.1 )
     [13]  Oracle Database                                ( ver. 1.1 )
   C [14]  PeopleSoft 8.x                                 ( ver. 1.1 )
     [15]  SAP Enterprise Portals                         ( ver. 1.1 )
   C [16]  Siebel 7.x                                     ( ver. 1.1 )
  *C [17]  SQL Server 7.5/2000                            ( ver. 1.2 )

--------------------------------------------
[Main Menu] Select an option:
    <L>ist current tuners
    <E>nable a tuner (or E#)
    <D>isable a tuner (or D#)
    <#> to configure a tuner
    <Q>uit
Select [L,E,D,Q,#] ? _
```

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.

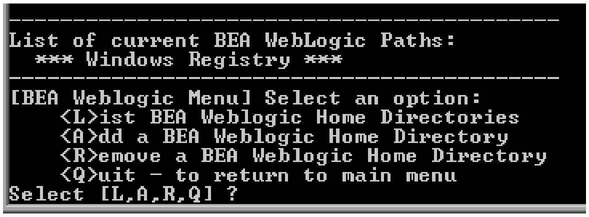
2 Type the tuner ID (the number in brackets that appears before the tuner's name) and press <Enter>.

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:



```
-------------------------------------------------
List of current BEA WebLogic Paths:
  *** Windows Registry ***
-------------------------------------------------
[BEA Weblogic Menu] Select an option:
    <L>ist BEA Weblogic Home Directories
    <A>dd a BEA Weblogic Home Directory
    <R>emove a BEA Weblogic Home Directory
    <Q>uit - to return to main menu
Select [L,A,R,Q] ?
```

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:

Use the Scheduled Tasks wizard to schedule pe_agent.exe to run when Windows starts up, as follows:

1. Select **Start** > **Settings** > **Control Panel** > **Scheduled Tasks** > **Add Scheduled Task**. The Scheduled Task Wizard opens.
2. Click **Next** and locate pe_agent.exe in the tuning agent's /bin directory.
3. In the dialog box where you specify when the task should be performed, choose "When my computer starts".
4. Enter the operating system username and password under which the task will run.
5. Click **Finish** to exit the wizard. pe_agent.exe is added to the list of scheduled tasks.
6. In the Scheduled Tasks window (**Start** > **Settings** > **Control Panel** > **Scheduled Tasks**), right-click the task created for pe_agent.exe , choose Properties, and click the Settings tab.
7. Clear the following boxes:
   - **Stop the task if it runs...**
   - **Don't start the task if the computer is running on batteries**
   - **Stop the task if battery mode begins**
8. Click **OK**.

On a UNIX system:

Update the rc.local script to launch pe_agent when UNIX starts up, as follows:

1. Locate the rc.local file. The location of the file depends on the operating system. For example, on a Linux system the file would be located in /etc/rc.d/rc.local.

2 Append the following calls to the rc.local file:

```
PE_USE_PORT=0

PE_USE_SSL=true

PE_HOME=/apps/tuningagent

/apps/tuningagent/agent/bin/pe_agent $PE_USE_PORT $PE_USE_SSL
$PE_HOME &
```

Note: The above script syntax is for BSH. Make sure that you run pe_agent in the background, using an ampersand (“&”).

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.

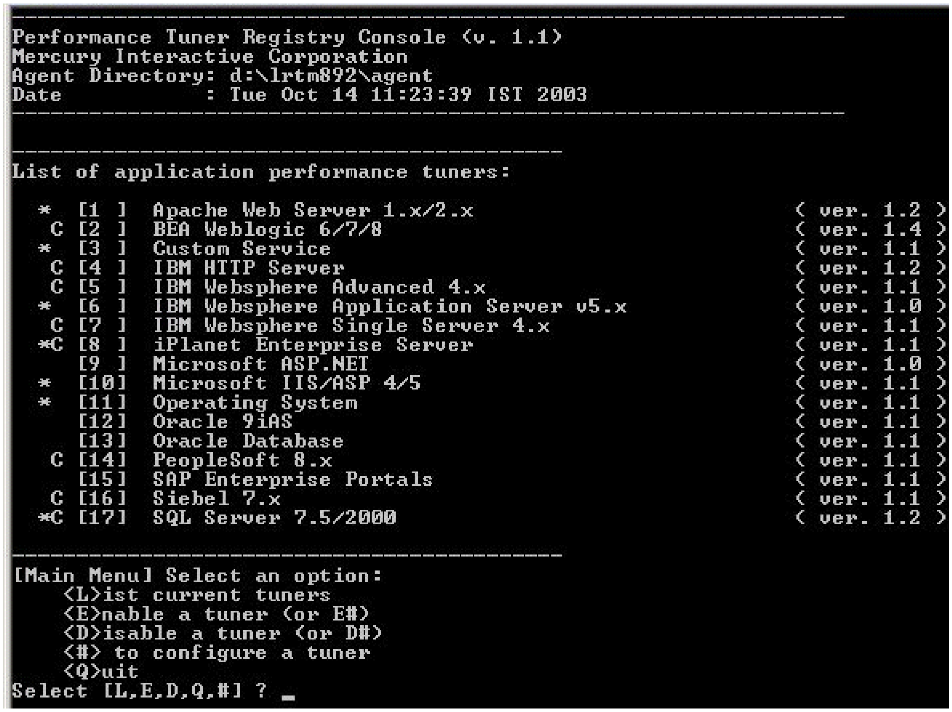
Configuring the Tuner

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:

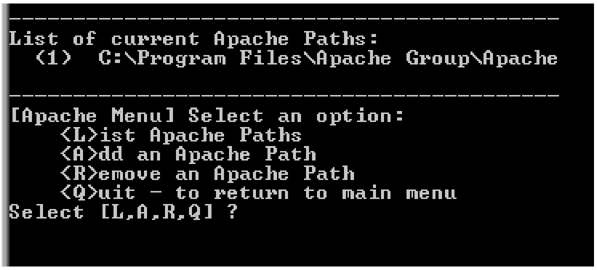
1 On the host machine, invoke the Performance Tuner Registry Console (see “Using the Performance Tuner Registry,” on page 438).

The list of tuners is displayed.



```
-----------------------------------------------------------------
Performance Tuner Registry Console (v. 1.1)
Mercury Interactive Corporation
Agent Directory: d:\lrtm892\agent
Date           : Tue Oct 14 11:23:39 IST 2003
-----------------------------------------------------------------

-------------------------------------------
List of application performance tuners:

  *  [1 ]  Apache Web Server 1.x/2.x                        ( ver. 1.2 )
   C [2 ]  BEA Weblogic 6/7/8                               ( ver. 1.4 )
  *  [3 ]  Custom Service                                   ( ver. 1.1 )
   C [4 ]  IBM HTTP Server                                  ( ver. 1.2 )
   C [5 ]  IBM Websphere Advanced 4.x                       ( ver. 1.1 )
  *  [6 ]  IBM Websphere Application Server v5.x            ( ver. 1.0 )
   C [7 ]  IBM Websphere Single Server 4.x                  ( ver. 1.1 )
  *C [8 ]  iPlanet Enterprise Server                        ( ver. 1.1 )
     [9 ]  Microsoft ASP.NET                                ( ver. 1.0 )
  *  [10]  Microsoft IIS/ASP 4/5                            ( ver. 1.1 )
  *  [11]  Operating System                                 ( ver. 1.1 )
     [12]  Oracle 9iAS                                      ( ver. 1.1 )
     [13]  Oracle Database                                  ( ver. 1.1 )
   C [14]  PeopleSoft 8.x                                   ( ver. 1.1 )
     [15]  SAP Enterprise Portals                           ( ver. 1.1 )
   C [16]  Siebel 7.x                                       ( ver. 1.1 )
  *C [17]  SQL Server 7.5/2000                              ( ver. 1.2 )

-------------------------------------------
[Main Menu] Select an option:
    <L>ist current tuners
    <E>nable a tuner (or E#)
    <D>isable a tuner (or D#)
    <#> to configure a tuner
    <Q>uit
Select [L,E,D,Q,#] ? _
```

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.



```
-------------------------------------------
List of current Apache Paths:
  (1)  C:\Program Files\Apache Group\Apache

-------------------------------------------
[Apache Menu] Select an option:
    <L>ist Apache Paths
    <A>dd an Apache Path
    <R>emove an Apache Path
    <Q>uit - to return to main menu
Select [L,A,R,Q] ?
```

The path must specify the directory that contains the Apache.exe file, and must **not** specify a subdirectory of that directory.

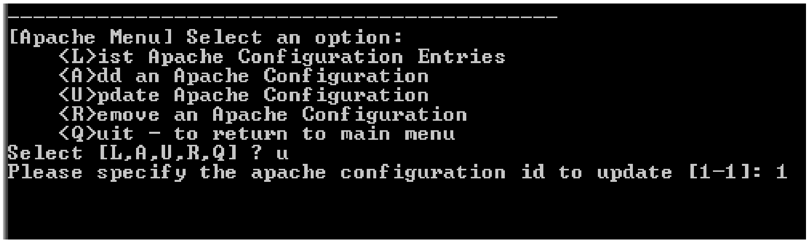
For example, for an Apache Web server installation on a Windows computer,
C:\Program Files\Apache Group\Apache is acceptable;
C:\Program Files\Apache Group\Apache**\bin** is not.

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:

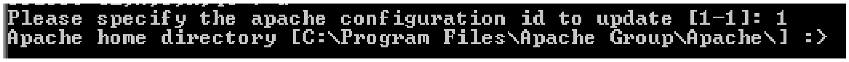
1 From the Web server configuration menu, type **U** to update the configuration and press <Enter>. The Performance Tuner Registry asks for the ID of the Web server that you are configuring. (This is the number to the left of the relevant path in the list of current Apache paths.)



```
-------------------------------------------
[Apache Menu] Select an option:
    <L>ist Apache Configuration Entries
    <A>dd an Apache Configuration
    <U>pdate Apache Configuration
    <R>emove an Apache Configuration
    <Q>uit - to return to main menu
Select [L,A,U,R,Q] ? u
Please specify the apache configuration id to update [1-1]: 1
```

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 446.



```
Please specify the apache configuration id to update [1-1]: 1
Apache home directory [C:\Program Files\Apache Group\Apache\] :>
```

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 443.

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 443.

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.

34

Configuration Tab Functions

This chapter describes various functions that are available to you via the Configuration tab.

The chapter describes the following functions:

- Starting a Service
- Stopping a Service
- Rebooting Host Machines
- Reconnecting the Console to a Server
- Stopping a Tuning Agent
- Printing Host Configurations
- Refreshing Service/Host Configuration
- Removing a Host from the 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, Mercury Tuning Console displays an error message.

For details of the various types of users and their usernames and passwords, see the table on page 396.

Starting 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**.

Mercury Tuning Console 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.

Stopping 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**.

Mercury Tuning Console 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.

Rebooting Host Machines

(For users with administrator access only).

Mercury Tuning Console 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 438 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 then choose **Reboot Host**.

Mercury Tuning Console 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**.

Mercury Tuning Console displays a dialog box requesting confirmation.

2 Click **Reboot All** to reboot all the host machines.

Reconnecting 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.

Stopping a Tuning Agent

To stop the tuning agent that is running on a server, click the server's icon and click the **Stop Tuning Agent** button. When Mercury Tuning Console asks you to confirm the action, click **Yes**.

Printing Host Configurations

You may find it useful to keep a hard-copy record of a host's tuning settings. Mercury Tuning Console 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.

Refreshing Service/Host Configuration

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**.

Removing a Host from the Server Configurations Tree

To remove a host from the Server Configurations tree, right-click the host's icon and choose **Remove**.

35

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 400.

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

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 VIII

Appendixes

A

Enabling Citrix Support

This chapter explains how to enable Citrix support in Mercury Tuning Console.

It includes the following section:

➤ Citrix Support Guidelines

Citrix Support Guidelines

To enable Citrix support, follow these guidelines:

➤ Install the launcher on the load generator machine as a process, not a service.

➤ When you define a Citrix load generator from the Console, use the load generator name together with an index, where the index range begins at 1 and ends at the number of terminals or Citrix hosts, for example:

remote1:1

remote1:2

remote1:3

➤ Modify the configuration file on the load generator machine located at launcher_service\dat\br_lnch_server.cfg and set the value of CitrixisActive in the Citrix section to 1.

B

Troubleshooting the Console

Mercury Tuning Console enables you to test entire applications. If one of the components of the application is not configured properly, Mercury Tuning Console sessions will not run.

This appendix discusses the most common Mercury Tuning Console problems:

- Mercury Tuning Console 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
- Console Exits Unexpectedly
- Cannot Open Topology Window or Run Execution Step
- Troubleshooting Remote Tuning

About Troubleshooting

Mercury Tuning Console 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.

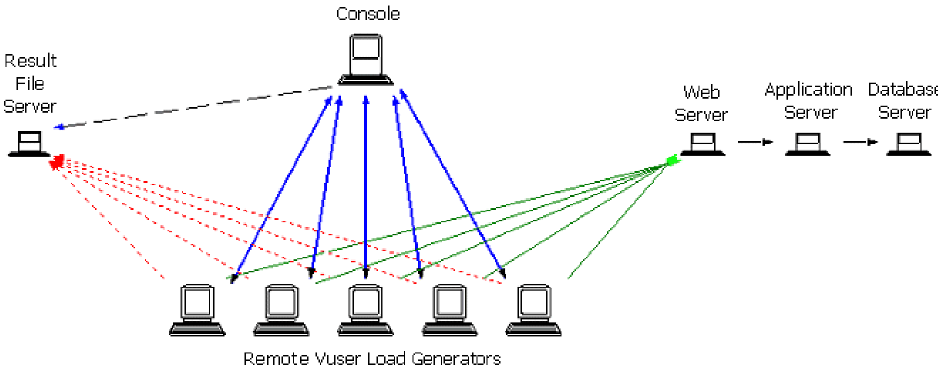
For more information on running scripts in stand-alone mode, refer to the appropriate guide for creating scripts.

Mercury Tuning Console Communications

Most communication problems can be solved if you understand your Mercury Tuning Console 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 Mercury Tuning Console. There are five servers: The Mercury Tuning 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.

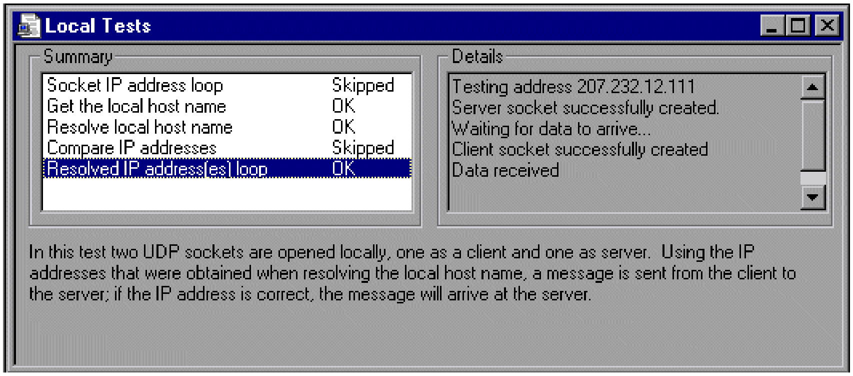
Failure to Communicate with a Load Generator

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. Mercury Tuning Console includes a utility called Hostinfo (hostinfo.exe), located under Mercury Tuning Console'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.



Local Tests

Summary

Socket IP address loop	Skipped
Get the local host name	OK
Resolve local host name	OK
Compare IP addresses	Skipped
Resolved IP address(es) loop	OK

Details

```
Testing address 207.232.12.111
Server socket successfully created.
Waiting for data to arrive...
Client socket successfully created
Data received
```

In this test two UDP sockets are opened locally, one as a client and one as server. Using the IP addresses that were obtained when resolving the local host name, a message is sent from the client to the server; if the IP address is correct, the message will arrive at the server.

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:

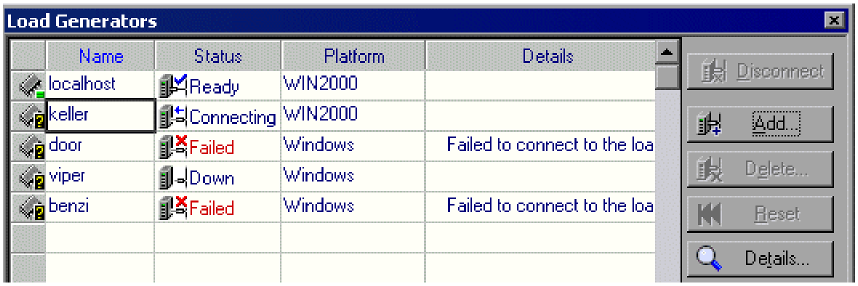
```
#     102.54.94.97     rhino.acme.com     # source server
#     38.25.63.10      x.acme.com         # x client host
```

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.)



Load Generators

Name	Status	Platform	Details
localhost	Ready	WIN2000	
keller	Connecting	WIN2000	
door	Failed	Windows	Failed to connect to the loa
viper	Down	Windows	
benzi	Failed	Windows	Failed to connect to the loa

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 115.

To use UNIX without RSH:

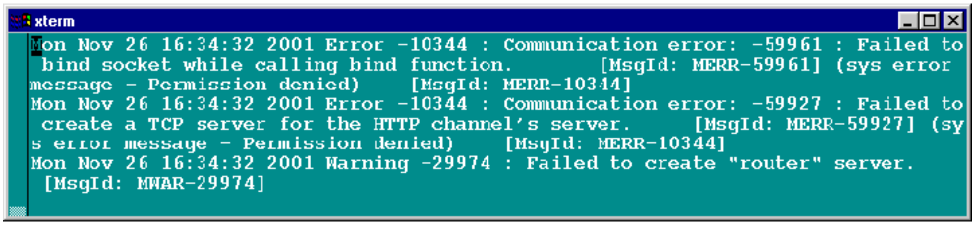
1 On the UNIX Load Generator machine, run the agent daemon by running the following command from *<Mercury Tuning Console directory>/bin*:

```
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 *<Mercury Tuning Console 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:



```
Mon Nov 26 16:34:32 2001 Error -10344 : Communication error: -59961 : Failed to
 bind socket while calling bind function.        [MsgId: MERR-59961] (sys error
message - Permission denied)     [MsgId: MERR-10344]
Mon Nov 26 16:34:32 2001 Error -10344 : Communication error: -59927 : Failed to
 create a TCP server for the HTTP channel's server.     [MsgId: MERR-59927] (sy
s error message - Permission denied)     [MsgId: MERR-10344]
Mon Nov 26 16:34:32 2001 Warning -29974 : Failed to create "router" server.
 [MsgId: MWAR-29974]
```

These messages appear because the Mercury Tuning Console 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 *<Mercury Tuning Console 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 Mercury Tuning Console 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, Mercury Tuning Console 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

Mercury Tuning Console 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:

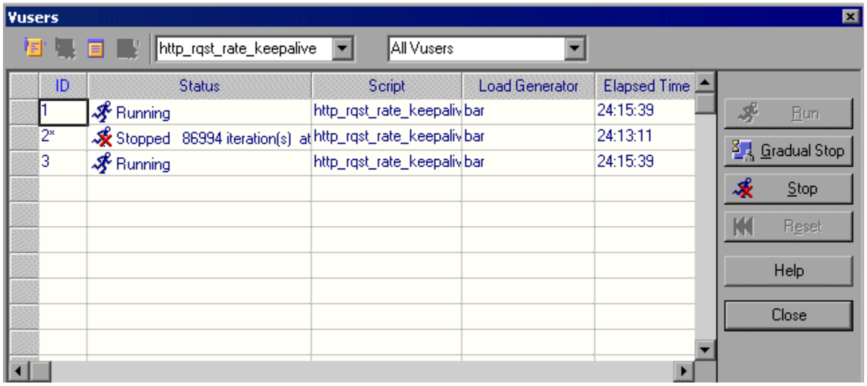
- Status Bar
- Output Window
- Output File (excluding GUI Vusers)
- Analysis Reports and Graphs

Status Bar

The status bar (located below the tool bar) provides a synopsis of the running step. It includes information on failed and passed transactions, as well as the step 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, Mercury Tuning Console displays an error status. If a Vuser completes the script execution, Mercury Tuning Console 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 18, "Running an Execution Step."



http_rqst_rate_keepalive

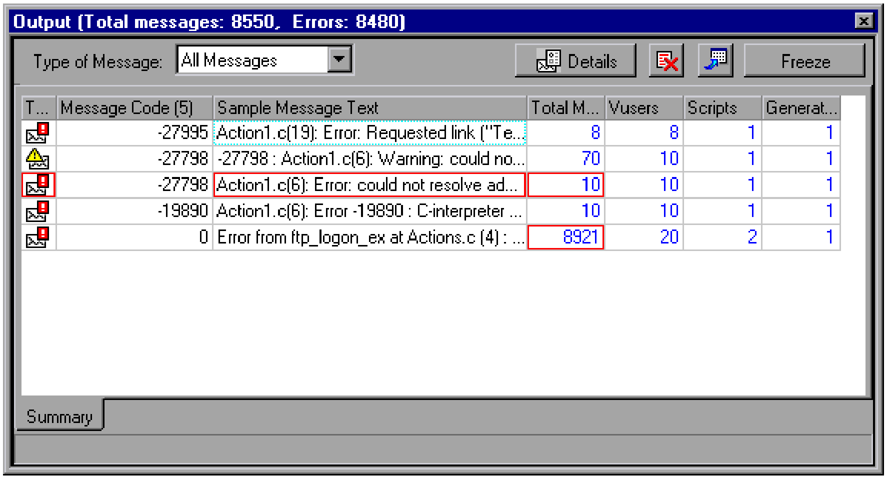
All Vusers

ID	Status	Script	Load Generator	Elapsed Time
1	Running	http_rqst_rate_keepaliv	bar	24:15:39
2*	Stopped 86994 iteration(s) at	http_rqst_rate_keepaliv	bar	24:13:11
3	Running	http_rqst_rate_keepaliv	bar	24:15:39

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.



Output (Total messages: 8550, Errors: 8480)

Type of Message: All Messages

T...	Message Code (5)	Sample Message Text	Total M...	Vusers	Scripts	Generat...
❗	-27995	Action1.c(19): Error: Requested link ("Te...	8	8	1	1
⚠	-27798	-27798 : Action1.c(6): Warning: could no...	70	10	1	1
❗	-27798	Action1.c(6): Error: could not resolve ad...	10	10	1	1
❗	-19890	Action1.c(6): Error -19890 : C-interpreter ...	10	10	1	1
❗	0	Error from ftp_logon_ex at Actions.c (4) : ...	8921	20	2	1

For more information about the Output window, see Chapter 19, "Viewing Vusers During Execution."

Output File

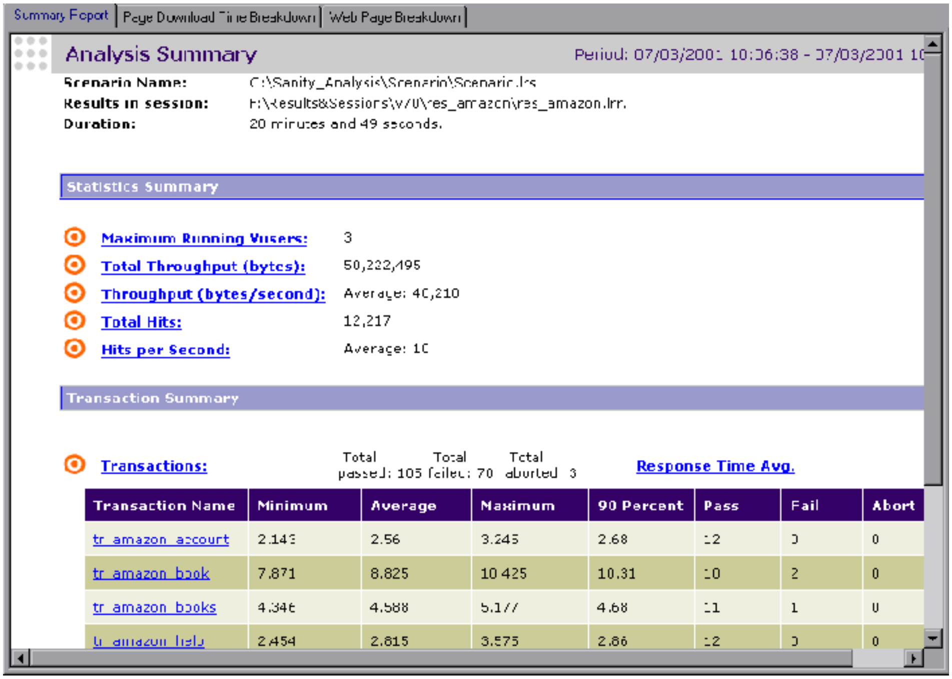
You can view information about script execution in an output file located in the Vuser result directory. The output file, *output.txt*, contains:

- 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 *Mercury Virtual User Generator User's 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.



Analysis Summary

Period: 07/03/2001 10:36:38 - 07/03/2001 10

Scenario Name: C:\Sanity_Analysis\Scenario\Scenario.lrs
Results in session: F:\Results&Sessions\v70\res_amazon\res_amazon.lrr
Duration: 20 minutes and 49 seconds.

Statistics Summary

Maximum Running Vusers:	3
Total Throughput (bytes):	50,222,495
Throughput (bytes/second):	Average: 40,210
Total Hits:	12,217
Hits per Second:	Average: 10

Transaction Summary

Transactions: Total passed: 105 Total failed: 70 Total aborted: 3 Response Time Avg.

Transaction Name	Minimum	Average	Maximum	90 Percent	Pass	Fail	Abort
tr_amazon_account	2.143	2.56	3.245	2.68	12	0	0
tr_amazon_book	7.871	8.825	10.425	10.31	10	2	0
tr_amazon_books	4.346	4.588	5.177	4.68	11	1	0
tr_amazon_help	2.454	2.815	3.575	2.86	12	0	0

For more information on the available graphs and reports, refer to the *LoadRunner 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.

Mercury Tuning Console 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 Mercury 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 Console'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 agent log file** contains information about communication activity between the Mercury Tuning Console agent and the MI Listener.

- ➤ To open the file on Windows machines, right-click the LoadRunner agent icon in the system tray of the LoadRunner agent machine, and select **View Log**. Alternatively, open the latest *<temp_directory>*\*<Mercury_Tuning_Console_agent_startup<unique identifier>.log* file (if the Mercury Tuning Console agent is a process), or *<temp_directory>*\Mercury_Tuning_Console_agent_service<unique identifier>.log file (if the Mercury Tuning Console 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.

The **MI Listener log file** contains information about MI Listener communication with the LoadRunner 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>*\Mercury_Tuning_Console_agent_startup<unique identifier>.log file (if the Mercury Tuning Console agent is a process), or *<temp_directory>\Mercury_Tuning_Console_agent_service<unique identifier>.log* file (if the Mercury Tuning Console agent is a service), in a text editor.

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 Mercury Tuning Console Agent and MI Listener

If there is a proper connection between the Mercury Tuning Console 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 Mercury Tuning Console agent machine. Local_machine_key is the value set in the Agent Configuration, as described in "Running Vusers Over a Firewall" on page 277 The file will be removed when the Mercury Tuning Console agent disconnects from the MI Listener.
- On both UNIX and Windows platforms, the following message will appear in the Mercury Tuning Console agent log file: Notify Connected to MI Listener.

Note: The Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console agent machine.
In both Windows and UNIX, the message Disconnected from MI Listener will appear in the Mercury Tuning Console agent log file.

UNIX Connection Errors

After installing the *m_agent_daemon* as described in Chapter 24, "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 *<Mercury_Tuning_Console_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.

Mercury Tuning Console 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 Mercury Tuning Console 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 F, "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 F, "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 Mercury Tuning Console agent machine (see Appendix F, "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 Mercury Tuning Console agent's CA List.
- The MI Listener's certificate has a future verification date.

See Appendix F, "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 Mercury Tuning Console agent's certificate does not have a signature that is included in the MI Listener's CA List.
- The Mercury Tuning Console agent's certificate has a future verification date.

See Appendix F, "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 Mercury Tuning Console agent's certificate has expired.

General Connection Errors

These errors can occur when using all configurations.

If no errors appear both in the Mercury Tuning Console 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 *<Mercury_Tuning_Console_Installation>\dat\br_lnch_server.cfg* file on the Mercury Tuning Console 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 Mercury Tuning Console 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 Mercury Tuning Console agent machine (e.g., by using the ping utility). If this fails, there is a configuration problem in the system not related to Mercury Tuning Console, 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).

Console Exits Unexpectedly

If the Console exits unexpectedly while running on a Windows NT 4 computer, ensure that you are not using more than one of the following monitors at the same time:

- Tuxedo
- Oracle
- SAP Client

Cannot Open Topology Window or Run Execution Step

When you try to open the Topology window or run an Execution step, the Console may display the message "Please wait. Monitors initializing". Subsequently the Topology window may fail to open or the step may not run.

If this happens, restart the remote registry service for the problematic monitor.

Execution Freezes after Closing Topology Window

When you close the Topology window, the Configuring Monitors icon may appear and remain on the task bar, and you may be prevented from continuing the session.

If this happens, restart the remote registry service for the problematic monitor.

Troubleshooting Remote Tuning

This section covers some issues you need to handle when using Mercury Tuning Console'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, Mercury Tuning Console 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.)

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.

Note: If you want to use the existing Java installation that came with an application (for example, with WebLogic), update the path so it points to the environment where the Java installation's jvm.dll file is located.

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 Mercury Tuning Console 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 Mercury Tuning 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.

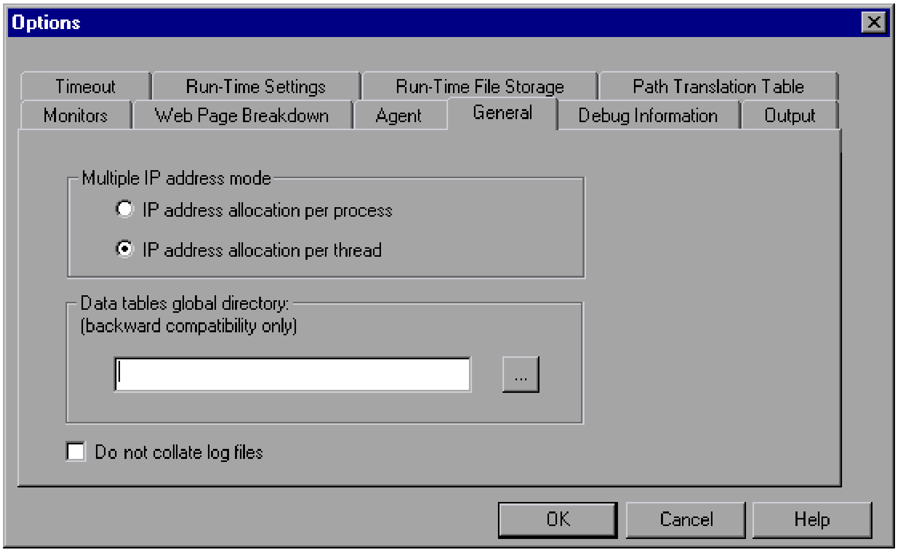
To exit the Expert mode, repeat the above process.

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 Mercury Tuning Console not to collate log files. This tab is displayed only when you operate the Console in the Expert Mode

To set the General Expert mode settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **General** tab.



Multiple IP address mode

○ IP address allocation per process

● IP address allocation per thread

Data tables global directory:
(backward compatibility only)

□ Do not collate log files

2 Select the Multiple IP address mode.

3 Enter the global directory for data tables.

4 If you want Mercury Tuning Console 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 Mercury Tuning Console.

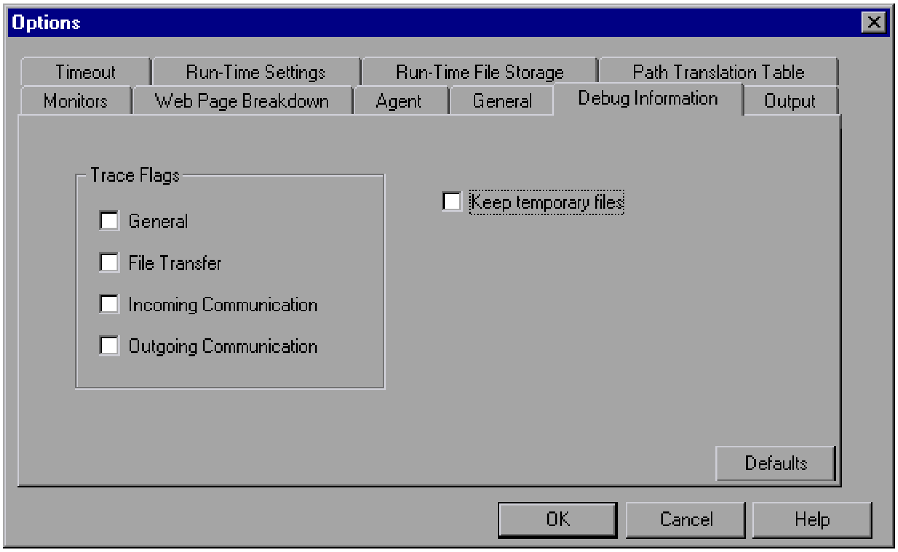
Do not collate log files: Instructs Mercury Tuning Console to collate only result files, and not log files.

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. This tab is displayed only when you operate the Console in the Expert Mode.

To set the Debug Information settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **Debug Information** tab.



Options

Trace Flags

□ General

□ File Transfer

□ Incoming Communication

□ Outgoing Communication

□ Keep temporary files

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 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, 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.

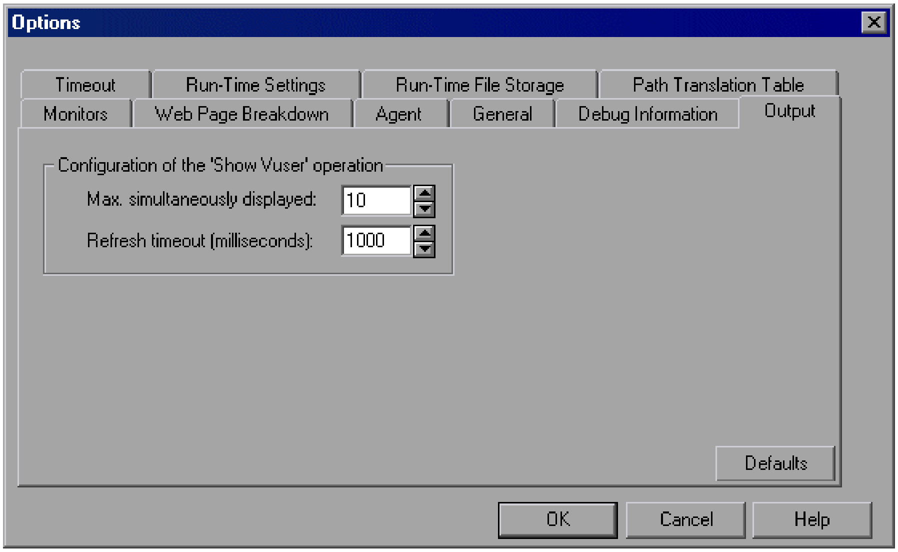
Options - Output Settings

Basic Output settings are available when Expert mode is disabled. However, Expert mode provides additional settings for handling Vuser output:

Save every xxx messages: Defines how often to auto-save the output message file. The default is every 100 messages.

To set the Output settings:

1 Choose **Tools** > **Options**. The Options dialog box appears. Select the **Output** tab.



Configuration of the 'Show Vuser' operation

Max. simultaneously displayed: 10

Refresh timeout (milliseconds): 1000

2 Enter the number of messages after which Mercury Tuning Console will perform an auto-save operation.

3 Enter values for the relevant Show Vuser options.

4 Click **OK** to accept the settings and close the dialog box.

Understanding the Options - Output Tab

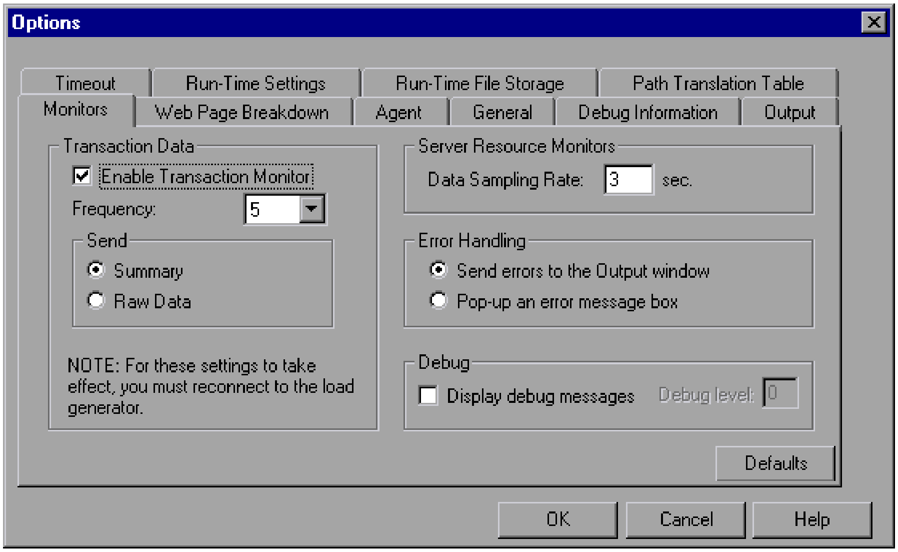
The Output tab enables you to configure the display of running Vusers on the Console machine.

- **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**.



Transaction Data

☑ Enable Transaction Monitor

Frequency: 5

Send

● Summary

○ Raw Data

NOTE: For these settings to take effect, you must reconnect to the load generator.

Server Resource Monitors

Data Sampling Rate: 3 sec.

Error Handling

● Send errors to the Output window

○ Pop-up an error message box

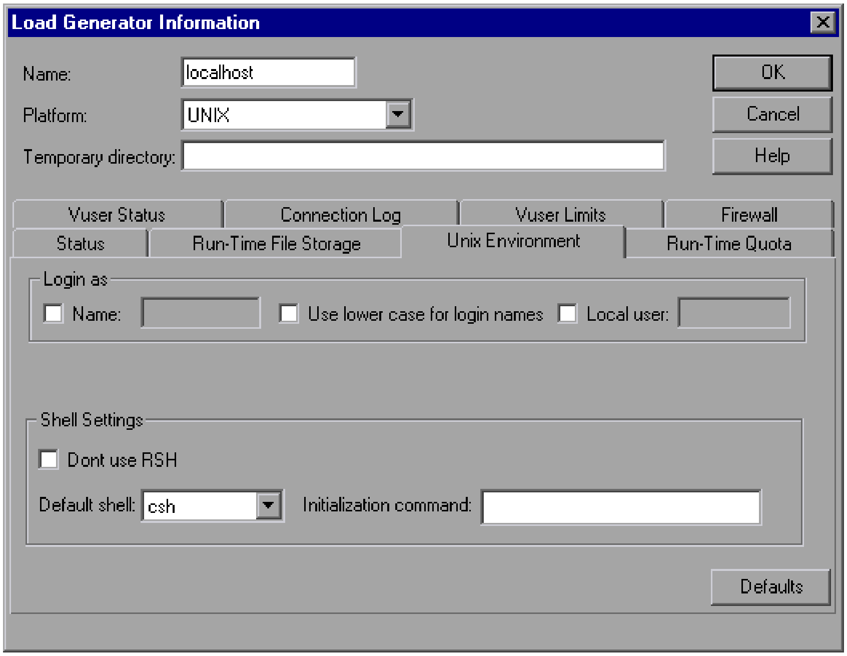
Debug

□ Display debug messages Debug level: 0

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

Name: localhost

Platform: UNIX

Temporary directory:

Login as

□ Name: □ Use lower case for login names □ Local user:

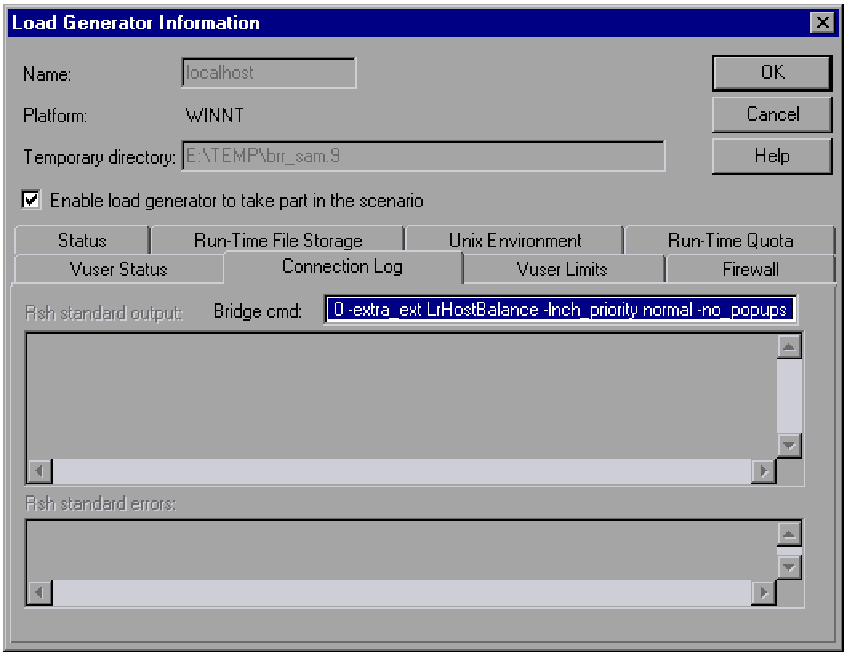
Shell Settings

□ Dont use RSH

Default shell: csh Initialization command:

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.



Load Generator Information

Name: localhost

Platform: WINNT

Temporary directory: E:\TEMP\brr_sam.9

☑ Enable load generator to take part in the scenario

Rsh standard output:

Bridge cmd: 0 -extra_ext LrHostBalance -lnch_priority normal -no_popups

Rsh standard errors:

Rsh standard output: Displays rsh standard output as the Console connects 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.

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, and in the Load Generator Information dialog box, select the **Connection Log** tab.
4. View the rsh standard output and rsh standard errors, or enter a new command in the Bridge cmd box to change the default bridge command.

D

Performing Path Translation

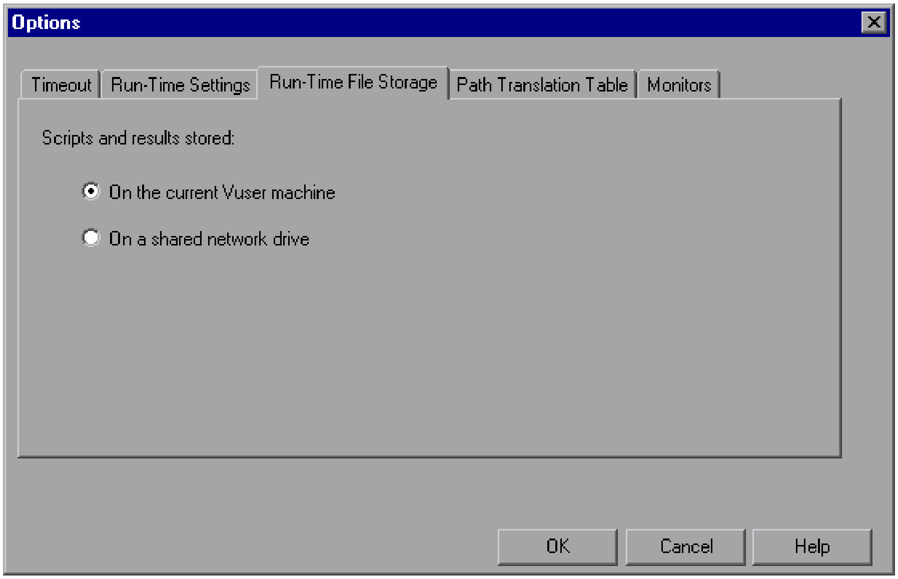
When you run a session step, Mercury Tuning Console gathers run-time data from the participating Vusers. By default, Mercury Tuning Console stores the data in temporary files on each Vuser machine. After the session step, the data is collated in the general results directory.

Alternatively, you can instruct Mercury Tuning Console 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 Mercury Tuning Console to convert a remote path name for the Console. A typical session step might have the Mercury Tuning 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 Mercury Tuning Console 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.



Scripts and results stored:

● On the current Vuser machine

○ On a shared network drive

Result and script files stored on a shared network drive require you to perform path translation.

The Session tab shows 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>]
```

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

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.

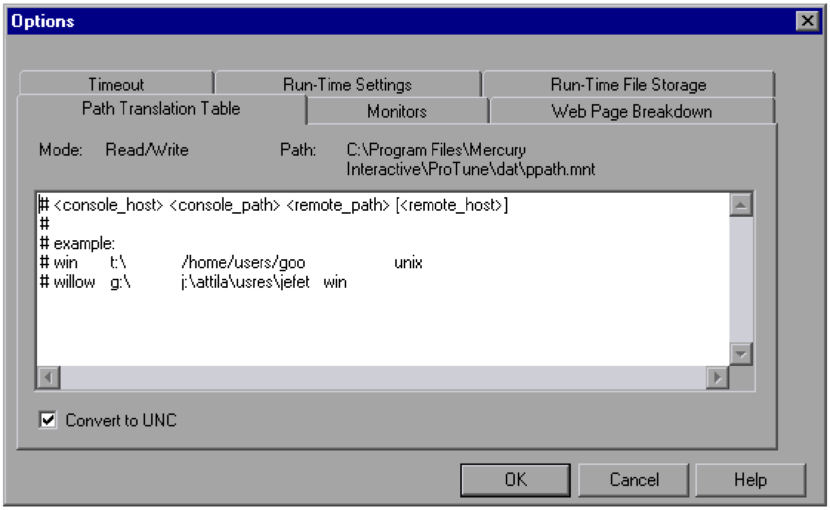
Editing the Path Translation Table

You maintain the Path Translation table using the Mercury Tuning Console. Mercury Tuning Console saves the Path Translation table as an ASCII file, **ppath.mnt.** This file, stored in Mercury Tuning Console_directory/dat, has a one–line entry for each network path to translate.

To edit the Path Translation table:

1 Start the Mercury Tuning Console.

2 Choose **Tools** > **Options** and select the **Path Translation Table** tab. The Path Translation Table view opens.



Mode: Read/Write

Path: C:\Program Files\Mercury Interactive\ProTune\dat\ppath.mnt

```
# <console_host> <console_path> <remote_path> [<remote_host>]
#
# example:
# win      t:\       /home/users/goo           unix
# willow   g:\       j:\attila\usres\jefet   win
```

☑ Convert to UNC

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 Mercury Tuning Console 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

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:

- About Multiple IP Addresses
- 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.

The 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:** Javascript Vuser, VB Vuser, VB Script Vuser
- **E-business:** FTP, Palm, SOAP, Web (HTTP/HTML), Web Services, WinSock\Web Dual Protocol
- **ERP/CRM:** Oracle NCA, Oracle Applications, PeopleSoft Enterprise, SAP-Web, Siebel-Web
- **Legacy:** RTE
- **Mailing Services:** Internet Messaging (IMAP), 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

Mercury 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 sessions.

The following procedure summarizes how to add new IP addresses to a load generator:

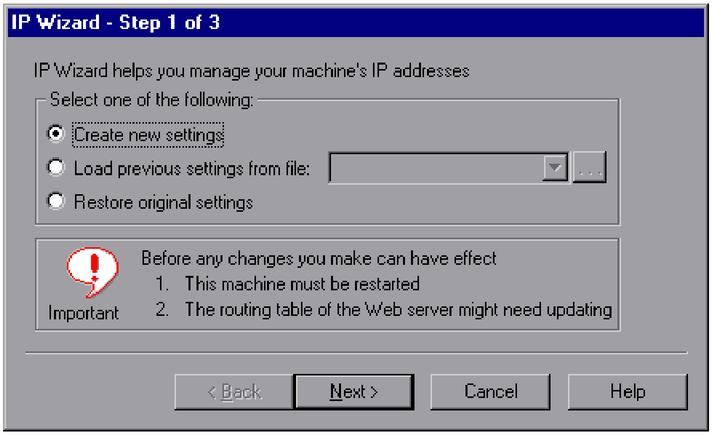
1. Run the IP Wizard on the load generator machine to add a specified number of IP addresses.
2. Restart the machine.
3. Update the server's routing table with the new addresses, if necessary.
4. Enable this feature from the Console. Refer to "Enabling Multiple IP Addressing from the Console" on page 521.

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 from the program group.



IP Wizard - Step 1 of 3

IP Wizard helps you manage your machine's IP addresses

Select one of the following:

● Create new settings

○ Load previous settings from file:

○ Restore original settings

Important

Before any changes you make can have effect

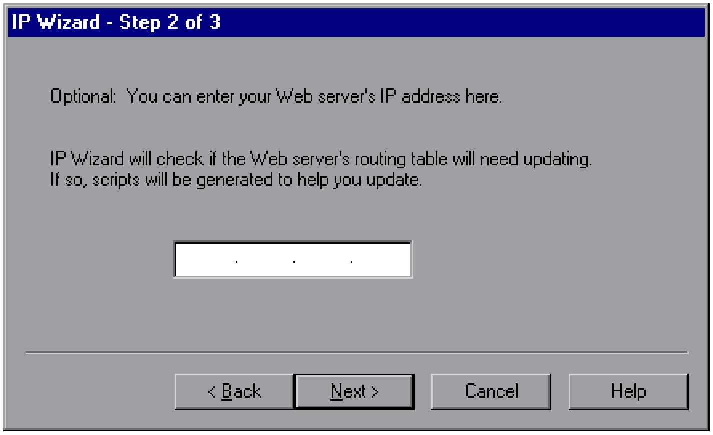
1. This machine must be restarted
2. The routing table of the Web server might need updating

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.



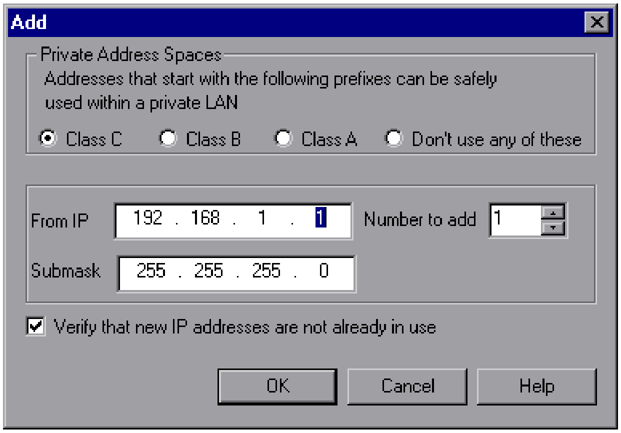
IP Wizard - Step 2 of 3

Optional: You can enter your Web server's IP address here.

IP Wizard will check if the Web server's routing table will need updating.
If so, scripts will be generated to help you update.

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 520 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.



Add

Private Address Spaces

Addresses that start with the following prefixes can be safely used within a private LAN

● Class C ○ Class B ○ Class A ○ Don't use any of these

From IP 192 . 168 . 1 . 1 Number to add 1

Submask 255 . 255 . 255 . 0

☑ Verify that new IP addresses are not already in use

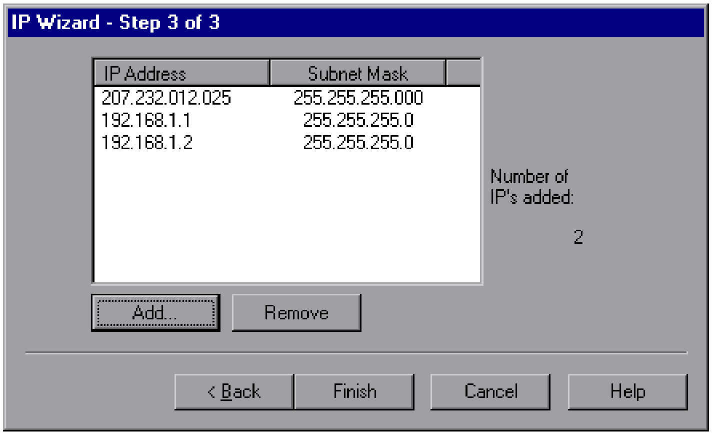
IP addresses include two components, a **netid** and **hostid**. The submask determines where the netid portion of the address stops and where the hostid begins.

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.

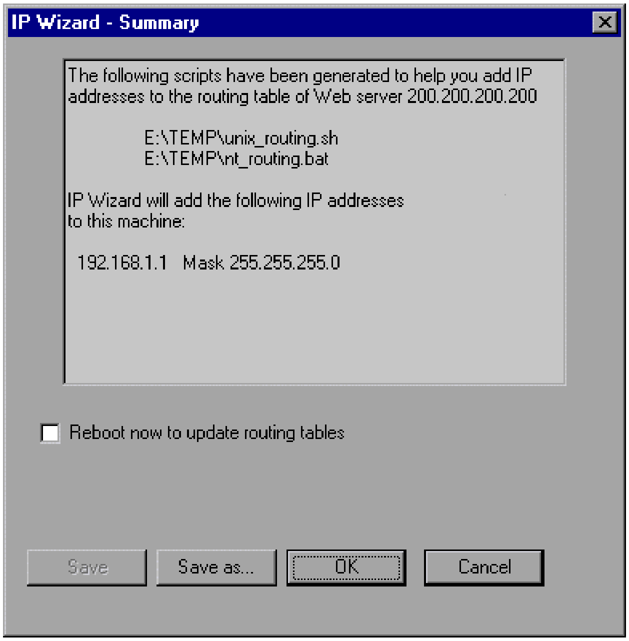
After the IP Wizard creates the new addresses, the summary dialog box lists all of the IP addresses.



IP Address	Subnet Mask
207.232.012.025	255.255.255.000
192.168.1.1	255.255.255.0
192.168.1.2	255.255.255.0

Number of IP's added: 2

10 Click **Finish** to exit the IP Wizard. The IP Wizard Summary dialog box is displayed.



The following scripts have been generated to help you add IP addresses to the routing table of Web server 200.200.200.200

E:\TEMP\unix_routing.sh
E:\TEMP\nt_routing.bat

IP Wizard will add the following IP addresses to this machine:

192.168.1.1 Mask 255.255.255.0

□ Reboot now to update routing tables

11 Note the address of the **.bat** file, and see “Updating the Routing Table” on page 520 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

➤ Linux

➤ HP 11.0 or higher

➤ IBM AIX

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.

IBM AIX

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:

```
/usr/sbin/ifconfig [int] [ip address] alias netmask [mask]
```

For example, if you want to add IP address 10.0.0.1 to the main interface, you need to run, as root, the following:

```
/usr/sbin/ifconfig ne0 10.0.0.1 alias netmask 255.255.255.0
```

To execute this line upon boot, create a standard script in the appropriate run level (/etc/rc.d/rc#.d).

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 Session tab, select **Session** > **Enable IP Spoofer**.

Note: You must select this option before connecting to a load generator.

---

2 In Expert mode (**Tools** > **Expert Mode**), use **Tools** > **Options** > **General** to specify how the Console should implement this feature.

For more information, refer to Appendix C, "Working in Expert Mode."

F

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:

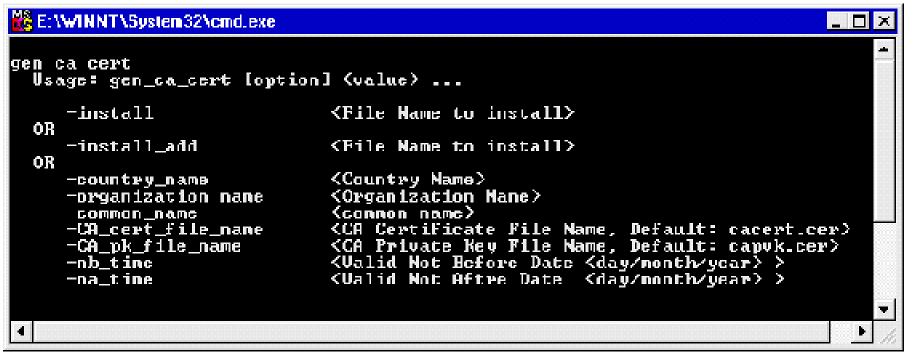
- Creating and Using Digital Certificates

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 create a Certificate Authority using gen_ca_cert:

1 To view the format and usage, run the **gen_ca_cert** utility from the <LoadRunner root folder>\launch_service\bin directory.



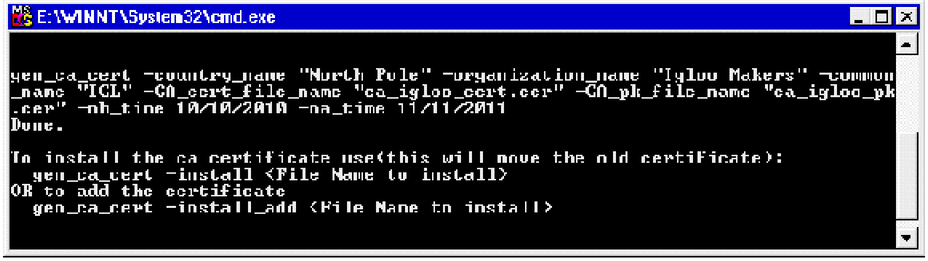
```
gen ca cert
  Usage: gen_ca_cert [option] <value> ...

     -install                 <File Name to install>
  OR
     -install_add             <File Name to install>
  OR
     -country_name            <Country Name>
     -organization name       <Organization Name>
      common_name             <common name>
     -CA_cert_file_name       <CA Certificate File Name, Default: cacert.cer>
     -CA_pk_file_name         <CA Private Key File Name, Default: capvk.cer>
     -nb_time                 <Valid Not Before Date <day/month/year> >
     -na_time                 <Valid Not Aftre Date  <day/month/year> >
```

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>.

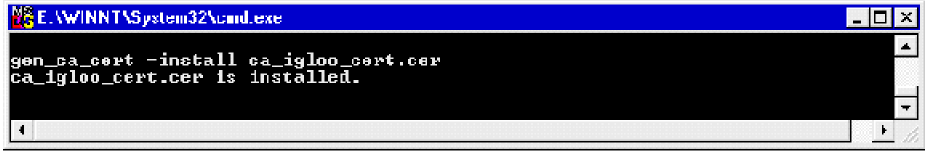
The following example creates two files: **ca_igloo_cert.cer** and **ca_igloo_pk.cer** in the current directory.:



```
gen_ca_cert -country_name "North Pole" -organization_name "Igloo Makers" -common
_name "ICL" -CA_cert_file_name "ca_igloo_cert.cer" -CA_pk_file_name "ca_igloo_pk
.cer" -nb_time 10/10/2010 -na_time 11/11/2011
Done.

To install the ca certificate use(this will move the old certificate):
  gen_ca_cert -install <File Name to install>
OR to add the certificate
  gen_ca_cert -install_add <File Name to install>
```

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>.

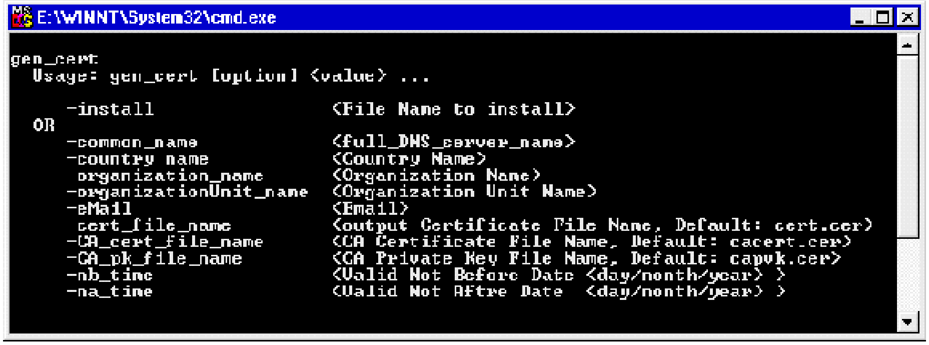


```
gen_ca_cert -install ca_igloo_cert.cer
ca_igloo_cert.cer is installed.
```

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 root folder>\launch_service\bin directory.



```
gen_cert
  Usage: gen_cert [option] <value> ...

     -install                  <File Name to install>
  OR
     -common_name              <full_DNS_server_name>
     -country name             <Country Name>
      organization_name        <Organization Name>
     -organizationUnit_name    <Organization Unit Name>
     -eMail                    <Email>
      cert_file_name           <output Certificate File Name, Default: cert.cer>
     -CA_cert_file_name        <CA Certificate File Name, Default: cacert.cer>
     -CA_pk_file_name          <CA Private Key File Name, Default: capvk.cer>
     -nb_time                  <Valid Not Before Date <day/month/year> >
     -na_time                  <Valid Not Aftre Date  <day/month/year> >
```

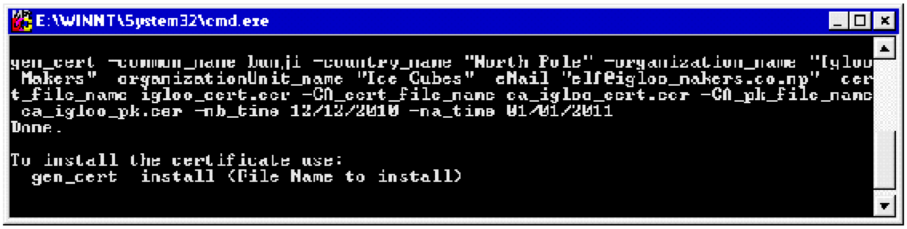
2 Create a new Digital Certificate by running the gen_cert command with at least one of the options: -country_name <country name>, -organization_name <organization name>, -organization_unit_name <organization unit name>, -eMail <email address> and -common_name <the name, full name or IP address of the 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:



```
gen_cert -common_name bunji -country_name "North Pole" -organization_name "Igloo
 Makers"  organizationUnit_name "Ice Cubes"  eMail "elf@igloo_makers.co.np"  cer
t_file_name igloo_cert.cer -CA_cert_file_name ca_igloo_cert.cer -CA_pk_file_name
 ca_igloo_pk.cer -nb_time 12/12/2010 -na_time 01/01/2011
Done.

To install the certificate use:
  gen_cert  install (File Name to install)
```

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.

Index

Host Resolution Functions Copyright Agreement